TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Nachrichtentechnik

Coding for Half-Duplex Relay Networks and The Trapdoor Channel

Tobias J. Lutz

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor–Ingenieurs

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr.-Ing. Norbert Hanik
Prüfer der Dissertation:	
1.	Univ.-Prof. Dr. sc. techn. Gerhard Kramer
2.	Prof. Frank R. Kschischang, Ph.D.
	University of Toronto, Canada

Acknowledgements

This dissertation is the result of my work as research and teaching assistant at the Institute of Communications Engineering of the Technische Universität München.

First of all, I want to thank Professor Gerhard Kramer for his support and guidance during my time at the Institute of Communications Engineering. His exceptional knowledge of the field and his enthusiasm for research were a great source of inspiration. Indeed, Gerhard's earlier research laid the cornerstone of the results presented in the first part of this thesis.

In 2011, I had the great pleasure of working with Professor Frank Kschischang from the University of Toronto. Frank's creative style of thinking truly influenced my own style of approaching research problems. I want to thank him for serving on my committee as well as for entertaining conversations. I still use the puzzle ball Frank gave me to confuse people.

My Ph.D. endeavor at the Institute of Communications Engineering would not have been possible without Professor Ralf Kötter, who died too soon. Ralf supervised my diploma thesis and offered me a Ph.D. position in 2008. Though I planned to do a Ph.D. in the U.S. at that time and already started preparing for the GRE test, I couldn't resist this offering after having made the experience to work with Ralf. My deepest thanks go to him.

I also want to thank Prof. Haim Permuter. The very first group meeting I attended at Stanford University was given by Haim. Haim talked about channels with memory, which awakenend my interest for the trapdoor channel. We chatted a little bit after the meeting and Haim suggested me to use a result from Ash's information theory book, which turned out to be very helpful.

During the first three years of my Ph.D., I collaborated with Dr. Christoph Hausl (now with Rhode & Schwarz). We were part of the European project N-Crave and attended

Contents

List of Figures

List of Tables

Abstract

Two information theoretic topics are studied: cascades/tree-structured networks of error-free half-duplex-constrained relays and the trapdoor channel. In the former case, a coding scheme is developed which represents data by an information-dependent allocation of the transmission slots of the relays. The strategy achieves capacity for a single source. For cascades/trees composed of an infinite number of half-duplex-constrained relays and a single source, an explicit capacity expression is derived. Interestingly, the capacity in bits per use in the binary case is equal to the logarithm of the golden ratio. If the cascades/trees include a source and relays with their own information, we show that the strategy achieves the cut-set bound when the rates of the relay sources fall below certain thresholds. Subsequently, we demonstrate that certain well-studied classes of codes, namely erasure codes and constrained codes, are natural codes in the context of half-duplex transmission. Specifically, we focus on noise-free half-duplex-constrained line networks with two sources where the first node and either the second node or the second-last node act as sources. In both cases, the capacity region is established. The achievability scheme for the first case uses the new idea that links with a half-duplex-constrained sink can be interpreted as erasure channels. We then address infinite-depth trees of error-free half-duplex-constrained relays and show that the multicast capacity is achieved using constrained coding at the source and simple symbol forwarding at the relays. All codes introduced in this thesis outperform the standard approaches for half-duplex networks, namely to require each network node to listen and to send half of the time, organized in a deterministic fashion. In the second part of this thesis, the problem of maximizing the n-letter mutual information of the trapdoor channel is considered. We show that $\frac{1}{2}\log_2\left(\frac{5}{2}\right) \approx 0.6610$ bits per use is an upper bound on its capacity. This upper bound proves that feedback increases the capacity of the trapdoor channel. We further present two novel views on the trapdoor channel. First, by deriving the underlying iterated function system (IFS), it is shown that the trapdoor channel with input blocks of length n can be regarded as the nth element of a sequence approximating a fractal. Second, an algorithm is presented that characterizes the trapdoor channel and resembles the recursion of generating all permutations of a given string.

Zusammenfassung

Diese Arbeit untersucht zwei Themen aus der Informationstheorie: Kaskaden bzw. Netze mit einer Baumtopologie, aufgebaut aus rauschfreien, half-duplex beschränkten Relais, und den Trapdoor Kanal. Ein Kodierungsschema wird eingeführt, welches Sendedaten durch die informationsabhängige Allokation von Sendezeitschlitzen repräsentiert. Die Strategie erreicht die Kapazität im Falle einer Quelle. Für Kaskaden bzw. Bäume, bestehend aus einer unendlichen Anzahl von Relais und einer Quelle, wird eine explizite Kapazitätsformel hergeleitet. Interessanterweise ist die Kapazität in Bits pro Kanalnutzung im binären Fall gleich dem Logarithmus des Goldenen Schnitts. Im Fall von Kaskaden bzw. Bäumen, aufgebaut aus einer Quelle und aus Relais mit eigener Sendeinformation, zeigen wir, dass die Kodierungsstrategie die Cut-Set Schranke erreicht, falls die Senderaten der Relais-Quellen unterhalb bestimmter Schranken liegen. Anschließend demonstrieren wir, dass die Auslöschungskodierung und die Constraint-Kodierung natürliche Kodierungsansätze für half-duplex beschränkte Netze sind. Im Speziellen befassen wir uns mit zwei Typen rauschfreier half-duplex Kaskaden mit zwei Quellen. In beiden Fällen wird die Kapazitätsregion hergeleitet. Das Erreichbarkeitsschema für einen der beiden Fälle basiert auf der neuen Idee, dass Links mit einer half-duplex beschränkten Senke im eigentlichen Sinne Auslöschungskanäle sind. Anschließend wenden wir uns unendlich tiefen Bäumen hin. Wir zeigen, dass Constraint-Kodierung an der Quelle und das Weiterleiten von Symbolen an den Relais die Multicast-Kapazität erreicht. Alle vorgestellten Kodierungsansätze schlagen hinsichtlich der Übertragungsrate die in der Praxis verwendeten deterministischen Ansätze. Im zweiten Teil der Arbeit beschäftigen wir uns mit dem Trapdoor Kanals. Wir zeigen, dass $\frac{1}{2}\log_2\left(\frac{5}{2}\right) \approx 0.6610$ Bits/Nutzung eine obere Schranke für die Kapazität darstellt. Diese Schranke zeigt, dass Rückkopplung die Kapazität des Trapdoor Kanals erhöht. Abschließend präsentieren wir zwei neue Sichtweisen. Durch das Herleiten des iterierten Funktionensystems wird zunächst gezeigt, dass der Trapdoor Kanal mit Eingängen der Länge n als n^{tes} Element einer Folge betrachtet werden kann, welche ein Fraktal approximiert. Anschließend wird ein Algorithmus präsentiert, welche den Trapdoor Kanal vollständig beschreibt und einer Rekursion ähnelt, die alle Permutationen einer gegebenen Sequenz erzeugt.

1

Introduction

1.1. Motivation and Overview

Today's *Information Age* is characterized by a significant and growing portion of mankind having instant and ubiquitous access to information. The possibility that nearly everybody can disseminate information globally has a tremendous impact on our social, economic and political life. Underlying these cultural developments are digital communication technologies and, in particular, wireless communication technologies. Most of the wireless technologies have in common that affordable radio nodes are *half-duplex-constrained*, i.e., a radio node is not able to receive a useful signal at the same time and over the same frequency band within which it is transmitting. As a consequence, transmission and reception times of each radio node in a given frequency band have to be strictly separated. In standard designs of wireless networks, separation is achieved by organizing transmission and reception periods in a *deterministic* fashion using *time-division duplex* (TDD) or *frequency-division duplex* (FDD). However, a predetermined transmission-reception schedule falls short of the maximum achievable rate since it ex-

cludes the possibility to encode a part of the data in the *signal timing*, i.e., in the transmission-reception schedule. This observation was made in [1].

Chapters 2, 3, 4 build upon [1] and [2] and are dedicated to the design and analysis of timing codes. Certain relay networks are considered, namely directed cascades and trees of *error-free* half-duplex-constrained relays. The term error-free means that network links do not distort the transmitted codewords by adding noise. Specifically, we assume that adjacent nodes in the network are connected by $(q + 1)$-ary noiseless links where the nodes can choose from a q-ary transmission alphabet or decide to be silent. The term *error-free* might be misleading since the relays in the network actually erase a certain portion of the received symbols due to the half-duplex constraint. Therefore, the half-duplex constraint can be regarded as introducing noise— an observation made in Chapter 3. Based on the approach of refraining from channel noise, we are able to identify and understand effects stemming from the half-duplex constraint and construct timing codes tailored to these effects. Moreover, we demonstrate that certain well-studied classes of codes, namely *erasure codes* and *constrained codes*, are natural in the context of half-duplex-constrained transmission. The proposed codes are optimal in most cases and always outperform the simplest possible coding approach used in practical applications, namely to require each network node to listen and to send half of the time, organized in a deterministic fashion.

The idea of timing has been considered in other disciplines like *computer networking* and *neural coding*. In computer networking, timing has been used as a method for constructing *covert channels*, which enable the secret transfer of information. Switching between different packet rates or timing of acknowledgments in an information dependent manner are two examples for encoding secret information by means of timing. A survey of techniques for creating, detecting and combating covert channels is given in [3]. Another discipline where timing plays an important role is *neural coding*, i.e., the science of how the nervous system encodes signals which are delivered by the sensory organs. It is known that sensory signals are converted into sequences of action potentials called *spike trains* in which sensory information is encoded through the timing of the spikes [4]. Ongoing research tries to answer the question of how *spike patterns* represent our perception of the world.

Chapters 5 and 6 are independent of Chapters 2, 3, 4. The focus is on the *trapdoor*

channel, a discrete binary channel with *memory*, which was introduced by David Blackwell in 1961 [5]. The term memory refers to the property that each channel output depends not only on the current input but also on past inputs and outputs. Another term used for the trapdoor channel is *finite state channel (FSC)* because the past inputs and outputs can be interpreted as the *state* of the channel. Many well-known examples of FSCs serve as models for *fading channels* and for *intersymbol interference*. Within these models, it is commonly assumed that the current output and the resulting state are statistically independent conditioned on the current input and the previous state [6, Chapter 4.6]. This assumption is not valid for the trapdoor channel, which is one reason why deriving its capacity seems to be challenging and is an open problem. The feedback capacity on the other hand is known [7]. In Chapter 5, we derive an upper bound on the capacity of the trapdoor channel. It turns out that this upper bound is strictly smaller than the feedback capacity, which shows that feedback increases the capacity of the trapdoor channel. This result is intuitive but not obvious at first glance. An example of a channel with memory where the no-feedback capacity equals the feedback capacity is the POST channel [8]. In Chapter 6, we investigate the structure of the trapdoor channel. We point out that its conditional probability matrix is a *fractal*. The approach is motivated by the idea that non-standard tools could help to solve the capacity problem. Moreover, we derive the underlying *iterated function system (IFS)*. Interestingly, the trapdoor channel is related to the Sierpinski triangle.

1.2. Outline and Contributions of the Thesis

This thesis is divided into 7 chapters. We briefly outline the contribution of each chapter below. The core of the material is presented in Chapters 2, 3, 4, 5, 6. The chapters are based on the following scientific articles:

▷ **T. Lutz**, C. Hausl, and R. Kötter, "Bits through deterministic relay cascades with half-duplex constraint," *IEEE Trans. Inf. Theory*, vol. 58, pp. 369–381, Jan. 2012.

▷ **T. Lutz**, G. Kramer, and C. Hausl, "Capacity for half-duplex line networks with two sources," in *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, Jun. 13-18

2010, pp. 2393–2397.

▷ F. R. Kschischang and **T. Lutz**, "A constrained coding approach to error-free half-duplex relay networks," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6258–6260, May 2013.

▷ **T. Lutz**, "Recursions for the trapdoor channel and an upper bound on its capacity," in Proc. *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, Jun. 29 - Jul. 4, 2014, pp. 2914–2918.

▷ **T. Lutz**, "Various views on the trapdoor channel and an upper bound on its capacity," submitted to *IEEE Trans. Inf. Theory*.

Chapter 2 continues our initial work started in [2]. We study error-free half-duplex-constrained relay cascades, i.e., networks where a source, a sink and a certain number of intermediate source/relay nodes are arranged on a line and adjacent node pairs are connected by error-free $(q + 1)$-ary pipes. A coding scheme is developed, which uses an information-dependent allocation of the transmission/reception slots of the relays. The basic idea of the scheme was already presented in [2]. We extend the scheme to q-ary transmission alphabets and present its recursive construction in an improved manner. We then show that the strategy achieves capacity for a single source and that the capacities are significantly higher than the rates achieved with a predetermined time-sharing approach. For cascades composed of an infinite number of half-duplex-constrained relays and a single source, we derive an explicit capacity expression. Interestingly, the capacity in bits/use for $q = 1$ is equal to the logarithm of the golden ratio. If the cascade includes a source and relays with their own information, we show that the strategy achieves the cut-set bound when the rates of the relay sources fall below certain thresholds. Finally, we point out that the coding scheme is well-suited for other half-duplex networks. This is demonstrated by means of broadcast trees and the butterfly network.

The focus of **Chapter 3** is on error-free half-duplex-constrained relay cascades with two sources, where the first node and either the second node or the second-last node in the cascade act as sources. In both cases, we establish the capacity region of rates at which both sources can transmit independent information to a common sink. The achievability scheme presented for the first case builds up on the scheme presented in

Chapter 2 and uses the new idea that links with a half-duplex-constrained sink can be interpreted as *erasure channels*. Therefore, the half-duplex constraint can be regarded as introducing noise. The achievability scheme for the second case is based on a random coding argument using superposition coding.

Chapter 4 shows that the multicast capacity of an infinite-depth tree-structured network of error-free half-duplex-constrained relays (which was derived in Chapter 2) can be achieved using constrained coding at the source and symbol forwarding at the relays. The approach is substantially less complex than the timing codes proposed in Chapter 2. Moreover, if applied to finite depth trees, the resulting transmission rates are strictly larger than the corresponding time-sharing rates. Finally, we provide examples of encoders and show how to construct them using the state-splitting algorithm.

In **Chapter 5**, the problem of maximizing the n-letter mutual information of the trapdoor channel is considered. We relax the problem by permitting distributions that are not probability distributions. Subsequently, we find explicit solutions for any $n \in \mathbb{N}$. It is then shown that $\frac{1}{2} \log_2 \left(\frac{5}{2} \right) \approx 0.6610$ bits per use is an upper bound on the capacity of the trapdoor channel. This upper bound, which is the tightest upper bound known, proves that feedback increases the capacity. As a byproduct, two recursions result, each of which is interesting in its own right.

In **Chapter 6**, two novel views are presented on the trapdoor channel. First, by deriving the underlying iterated function system (IFS), it is shown that the trapdoor channel with input blocks of length n can be regarded as the nth element of a sequence of shapes approximating a fractal. Second, an algorithm is presented that fully characterizes the trapdoor channel and resembles the recursion of generating all permutations of a given string.

In **Chapter 7**, we summarize and discuss open problems.

We adopt notational conventions that are standard. Random variables are denoted by capital letters (e.g., X) and their corresponding alphabets are denoted by corresponding calligraphic letters (e.g., $\mathcal{X}$). Frequently, random variables will appear with two subscripts (e.g., $X_{v,k}$). In this case, we are referring to the kth instance of the random vector $\mathbf{X}_v$. In later chapters, subindices indicate the dimension of vectors and matrices. They have to be taken to the power of two. E.g., $\mathbf{I}_n$ is the $2^n \times 2^n$ identity matrix. A list with the complete mathematical notation and abbreviations is shown in Appendix A.

2

Bits Through Deterministic Relay Cascades With Half-Duplex Constraint[*]

2.1. Introduction

Consider a relay cascade, i.e., a network where a source node, a sink node and a certain number of intermediate source/relay nodes are arranged on a line and where adjacent node pairs are connected by error-free $(q + 1)$-ary pipes. Suppose the source and a subset of the relays wish to communicate independent information to the sink under the condition that each relay in the cascade is *half-duplex-constrained*, i.e., is not able to transmit and receive simultaneously. This approach lets us understand half-duplex-constrained transmission without having to consider channel noise. Moreover, we may

---

[*]This chapter is based on the following publication [9]: *IEEE Trans. Inf. Theory*, vol. 58, no. 1, pp. 369-381, Jan. 2012 (together with Christoph Hausl and Ralf Kötter).
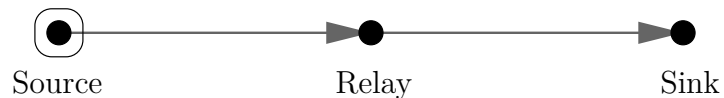
Figure 2.1.: A noiseless relay cascade composed of three nodes.

use combinatorial arguments instead of stochastic arguments. A natural strategy for half-duplex devices is to define a *time-division schedule* a priori. Under this assumption, the capacity or rate region of various half-duplex-constrained relay channels [10], [11] and networks [12] has been determined. We will, however, show that predetermined time-sharing falls considerably short of the theoretical optimum, i.e., higher rates are possible by an information-dependent allocation of the transmission/reception slots of the relays. The meaning of *information-dependent allocation* is illustrated in the following example.

Example 2.1. Consider the noiseless relay cascade depicted in Fig. 2.1. Let $\mathcal{W}_0 = \{0, \ldots, 7\}$ be the message set. In each block $b = 1, 2, \ldots$ of length 4, the source wishes to communicate a randomly chosen message $w_{0;b} \in \mathcal{W}_0$ to the sink via a single half-duplex-constrained relay node. A direct link between the source and the sink does not exist. Suppose the alphabet of both source and relay equals $\{0, 1, \mathsf{N}\}$ where N indicates a channel use without transmission and $\{0, 1\}$ is a $q = 2$-ary transmission alphabet. The half-duplex constraint is modeled as follows. When the relay uses symbol N, i.e., the relay is off, it is able to listen to the source without error and otherwise not. Let $\mathbf{x}_0(b)$ be the codeword chosen by the source encoder to represent $w_{0;b}$ in block b and let $\mathbf{x}_1(b)$ indicate the codeword chosen by the relay encoder for representing $w_{0;b-1}$ in block b. The coding scheme is illustrated in Table 2.1. The source encoder maps each message $w_{0;b}$ to $\mathbf{x}_0(b)$ by allocating the corresponding binary representation of $w_{0;b}$, i.e., three bits, to four time slots. The precise allocation of the three bits to four time slots is determined by the following protocol. In the first block, the source allocates three bits to the first three time slots of $\mathbf{x}_0(1)$. Now assume that the source has already sent codeword $\mathbf{x}_0(b)$ to the relay. Based on the first two binary digits of the noiselessly received codeword $\mathbf{x}_0(b)$, the relay encoder determines which of the four time slots to use for transmission in $\mathbf{x}_1(b+1)$. In particular, 00, 01, 10, 11 in $\mathbf{x}_0(b)$ tell the relay to send either in the first, the second, the third or the fourth time slot of $\mathbf{x}_1(b+1)$. The binary value to be transmitted in $\mathbf{x}_1(b+1)$ is equal to the third bit in $\mathbf{x}_0(b)$. Since the source encoder knows

Table 2.1.: Illustration of the Timing Scheme described in Example 2.1.

block b	$w_{0;b}$	$\mathbf{x}_0(b)$	$\mathbf{x}_1(b)$	$\hat{w}_{0;b-1}$
$b=1$	1 (001)	001N	NNNN	-
$b=2$	2 (010)	N010	1NNN	1
$b=3$	4 (100)	1N00	N0NN	2
$b=4$	7 (111)	11N1	NN0N	4
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

the scheme used by the relay, it can allocate its three new bits in $\mathbf{x}_0(b+1)$ to those slots in which the relay is able to listen. Hence, the relay encodes a part of its information in the timing of the transmission symbols. The sink forms the estimate $\hat{w}_{0;b-1}$ of $w_{0;b-1}$ based on the received relay codeword $\mathbf{x}_1(b)$ using both the position of the transmission symbol and its value. In this example, a rate of 0.75 bit per use (abbreviated as b/u in the remainder) is achievable if the number of blocks becomes large. By allowing arbitrarily long codewords, we will show that an extension of the strategy approaches 1.1389 b/u. A similar example for $q = 1$ was shown in [1, 13].

The outline of this chapter is as follows. Section 2.2 reviews information theoretic results about line networks. In Section 2.3, we introduce the channel model. A coding strategy is introduced in Section 2.4, which uses an information-dependent allocation of the transmission/reception slots of the relays. It is then shown in Section 2.5 that the strategy achieves the capacity of relay cascades with a single source. In particular, it approaches a rate equal to

$$C_{m-1}(q) \overset{def}{=} \max_{P_{X_0 \cdots X_m}} \min_{1 \leq i \leq m} H(Y_i|X_i), \tag{2.1}$$

where $m - 1$ indicates the number of relays in the cascade and X_i and Y_i are the sent and received symbol of the ith relay. For cascades composed of an infinite number of half-duplex-constrained relays, we show that the capacity is given by

$$C_{\infty}(q) \overset{def}{=} \log_2 \left(\frac{1 + \sqrt{4q+1}}{2} \right) \text{ b/u.}$$

Remarkably, $C_\infty(1)$ is equal to the logarithm of the golden ratio and $C_\infty(2)$ is 1 b/u. If the cascade includes a source and relays with their own information, the strategy achieves the cut-set bound if the rates of the relay sources fall below certain thresholds. This is shown in Section 2.6. We finally show in Section 2.7 that the proposed coding strategy can be applied to tree structured networks and to the half-duplex-constrained butterfly network. In the latter case, the proposed timing strategy outperforms the well-known XOR-based network coding strategy [14].

2.2. Related Literature

The classical relay channel goes back to van der Meulen [15]. Further significant results concerning capacity and coding were obtained by Cover and El Gamal in [16]. A comprehensive literature survey of various *decode-and-forward (DF)* and *compress-and-forward (CF)* strategies for small relay networks is given in [17]. General relay networks are difficult to analyze (even the capacity of the non-degraded single relay channel is an open problem). Motivated by the observation that cascades are more accessible for analysis and are fundamental building blocks of general communications systems, various source and channel coding problems have been examined (without the assumption of half-duplex-constrained nodes).

Yamamoto [18] considers a deterministic three node line network where the first node generates two random sequences. The region of achievable rates is found such that the second node is able to reconstruct the first sequence and the third node the second sequence within prescribed distortion tolerances. These results are extended to longer lines and branching communication systems in the same paper. A related version of the three node source coding problem is investigated in [19]. The encoder at the first node intends to communicate a random sequence within certain distortion constraints to the relay and the destination under the assumption that the relay and the destination have access to individual side information about the source. The authors derive inner and outer bounds for the rate-distortion region and characterize scenarios where both bounds coincide. Another distributed source coding problem for the three node line network is examined in [20]. In contrast to the cases before, the relay acts as a source which is

correlated to the source at the first node. The destination wishes to estimate a function of the output of the two sources. Inner and outer bounds on the achievable rate region are provided such that an arbitrarily chosen distortion constraint is satisfied.

The channel capacity of three node line networks with no processing at the middle terminal was examined in an early work [21]. The author asks which channel of the set of binary channels with equal capacity has to be cascaded with itself in order to achieve the largest end-to-end capacity. The answer is that a symmetric binary channel under cascade has a higher end-to-end capacity than an asymmetric binary channel under cascade with the same capacity (unless the channels have very low capacity). Finite length cascades of identical discrete memoryless channels are considered in [22] again under the assumption that the intermediate terminals do not possess any processing capability and that the transition matrix of the subchannels is nonsingular. By means of the eigenvalue decomposition of the transition matrix, the channel capacity is derived. Another work which investigates cascades composed of identical discrete memoryless channels is [23]. In contrast to [22], it is assumed that the intermediate relay nodes are able to process blocks of a fixed length. It is then shown that the capacity of the infinite length cascade equals the rate of the zero-error code of a single link and that the capacity is always upper-bounded by the zero-error capacity of a single link. In [24], the problem of finding the optimal ordering of a set of n distinct binary channels is analyzed such that the capacity of the resulting cascade is maximized. The question results from the observation that ordering has a strong influence on the capacity because matrix multiplication is not commutative. In the case of binary channels with positive determinants the authors are able to specify the optimal ordering.

In this chapter, the idea of timing is applied to half-duplex line networks. Timing is not a new idea in information theory and has already been used within the framework of queuing channels. Anantharam and Verdú showed [25] that encoding information into the time differences of arrivals to the queue achieves the capacity of the single server queue with exponential service distribution. The discrete-time version of this problem was analyzed in [26]. In [27], Kramer developed a memoryless half-duplex relay channel model and noticed that higher rates are possible when the transmission and reception time slots of the relay are random since one can send information through the timing of operation modes. The work in [1] further points out that half-duplex relay cascades

are physically degraded relay networks and hence their capacity with one message is achieved by the DF protocol derived in [28].

2.3. Network Model

Consider the discrete relay cascade as depicted in Fig. 2.2. The underlying topology corresponds to a directed path graph in which each node is labeled by a distinct number from $\mathcal{V} \stackrel{def}{=} \{0, \ldots, m\}$ with $m > 1$. The integers 0 and m belong to the source and the sink, respectively, while all remaining integers 1 to $m - 1$ represent half-duplex-constrained relays, i.e., relays which cannot transmit and receive at the same time. The connectivity within the network is described by the set of edges $\mathcal{E} \stackrel{def}{=} \{(i, i + 1) : 0 \leq i \leq m - 1\}$, i.e., the ordered pair $(i, i + 1)$ represents the communications link from node i to node $i + 1$. The output of the ith node, which is the input to channel $(i, i + 1)$ is denoted as X_i and takes values in the alphabet $\mathcal{X} \stackrel{def}{=} \{0, \ldots, q - 1\} \cup \{\mathsf{N}\}$ where $\mathcal{Q} \stackrel{def}{=} \{0, \ldots, q - 1\}$ denotes the q-ary transmission alphabet while N is meant to signify a channel use in which node i is not transmitting. We assume that all nodes use the same transmission alphabet (though the following derivations would hold if the cardinality of the transmission alphabets depends on the node index). The input of the ith node, which is the output of channel $(i - 1, i)$ is denoted as Y_i and is given by

$$Y_i = \begin{cases} X_{i-1}, & \text{if } X_i = \mathsf{N} \\ X_i, & \text{if } X_i \in \mathcal{Q} \end{cases}, \tag{2.2}$$

where $1 \leq i \leq m$. Channel model (2.2) captures the half-duplex constraint as follows. Assume relay i is in transmission mode, i.e., $X_i \in \mathcal{Q}$. Then relay i hears itself (i.e., $Y_i = X_i$) but cannot listen to node $i - 1$ or, equivalently, relay i and node $i - 1$ are disconnected. If relay i is not transmitting, i.e., $X_i = \mathsf{N}$, it is able to listen to relay $i - 1$ via a noise-free $(q+1)$-ary pipe (i.e., $Y_i = X_{i-1}$). The sink listens all the time. Hence, its input Y_m is given by $Y_m = X_{m-1}$. Another interpretation of the channel model is that the output X_i of relay i controls the position of a switch which is placed at its input. If relay i is transmitting, the switch is in position 1 otherwise it is in position 2 (see Fig. 2.2). Since a pair of nodes is either perfectly connected or disconnected, we obtain a deterministic
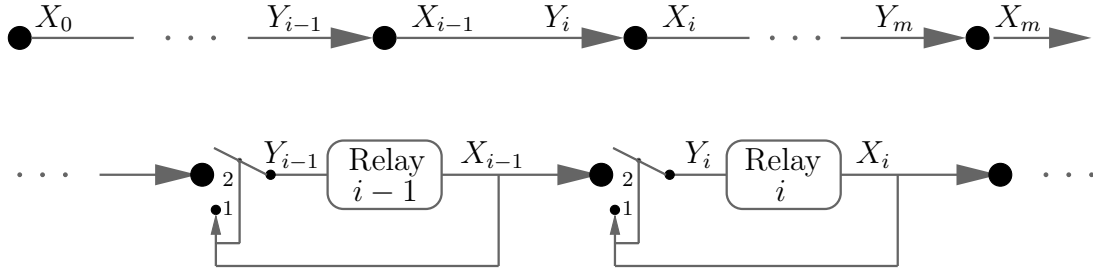
Figure 2.2.: A noiseless half-duplex relay cascade described by switches and feedback.

network with $P_{Y_1 \ldots Y_m | X_0 \ldots X_m} \in \{0, 1\}$ that factors as $\left[\prod_{i=1}^{m-1} P_{Y_i | X_i X_{i-1}} \right] P_{Y_m | X_{m-1}}$ where $P_{Y_i | X_i X_{i-1}}$ is determined by (2.2). Finally, we note that

$$X_{i,n} = f_i \left(\mathbf{Y}_i^{n-1}, W_i \right). \tag{2.3}$$

That is, the transmission symbol of node i at time n is a function $f_i(\cdot)$ of the previously received symbols $\mathbf{Y}_i^{n-1}$ and its message W_i.

2.4. A Timing Code for Relay Cascades

2.4.1. Information Flow

Every node $v \in \{0, \ldots, m-1\}$ draws its message W_v uniformly and independently from the message set $\mathcal{W}_v = \left\{ 1, \ldots, 2^{nR_v} \right\}$ where $w_{v;b}$ denotes the message sent by node v to node $v + 1$ in block b. Each block has length n. Observe that this setup includes the special case that only a subset of the relays communicate own information to the sink by setting the rate of the remaining relays to zero. The relays allocate their information to the codewords as follows. At the end of block $b - 1$, relay v carries out two tasks. It decodes the messages $\{w_{0;b-v}, \ldots, w_{v-1;b-1}\}$ from the received codeword $\mathbf{y}_v(b - 1)$ and draws a new message $w_{v;b}$. The decoded messages together with the new message are then forwarded to node $v + 1$ by choosing a sequence $\mathbf{x}_v(b)$. Source node 0 sends one message $w_{0;b}$ per block represented through $\mathbf{x}_0(b)$. We assume an initialization period of $m - 1$ blocks. In the first block node 0 forwards information, in the second block nodes 0 and 1 forward information and so forth. From the mth block onwards all nodes (except

the sink) forward information. Thus, the sink does not decode until the end of the mth block. Since a very large number of transmission blocks is considered, we can neglect the initial delay.

2.4.2. Construction

A coding strategy is introduced, which implements the information flow described in Section 2.4.1. The strategy relies on the observation that information can be represented not only by the value of code symbols but also by the position of code symbols, i.e., by timing the transmission and reception slots of the relay nodes. The strategy requires synchronization on the symbol level through a shared clock. The codebook construction is recursive and guarantees that adjacent nodes do not transmit at the same time. The following encoding techniques are applied at the source and the relays where n_v denotes the number of transmitted symbols of node v within one block of n symbols.

> ▷ At relay $m-1$: Relay $m-1$ represents information by choosing n_{m-1} transmission symbols per block from the q-ary transmission alphabet $\mathcal{Q}$ combined with allocating the n_{m-1} symbols to the transmission block of n symbols. Thus, $q^{n_{m-1}}\binom{n}{n_{m-1}}$ different sequences $\mathbf{x}_{m-1}$ of length n are available at relay $m-1$. Observe that $q^{n_{m-1}}$ equals the number of distinct sequences when the q-ary symbols are located at fixed slots while $\binom{n}{n_{m-1}}$ equals the number of possible transmission patterns.

> ▷ At relay v, $1 \leq v \leq m-2$: Observe that the effective codeword length of relay v reduces to $n-n_{v+1}$ since relay $v+1$ cannot listen to relay v when it (i.e., relay $v+1$) transmits. For each transmission-listen pattern used by node $v+1$, node v generates $q^{n_v}\binom{n-n_{v+1}}{n_v}$ different sequences by allocating n_v transmission symbols from the alphabet $\mathcal{Q}$ in all possible ways to the $n-n_{v+1}$ listen slots of the pattern. The remaining slots of the pattern, i.e., the slots in which node $v+1$ transmits, are filled with off symbols N.[1] Similar to before, q^{n_v} equals the number of possible distinct sequences when the q-ary symbols are located at fixed slots while $\binom{n-n_{v+1}}{n_v}$ equals the number of possible transmission-listen patterns of relay v given a particular pattern of relay $v+1$.

---

[1] Any symbols from $\mathcal{X}$ can be used for filling since the next relay is not able to listen.

▷ At node 0: The source uses the $(q+1)$-ary alphabet $\mathcal{X} = \mathcal{Q} \cup \{\mathsf{N}\}$ for encoding where the off symbol N is used as a regular alphabet symbol. Due to the half-duplex constraint at relay 1, the effective codeword length of the source reduces to $n - n_1$ since relay 1 cannot pay attention to the source when it (i.e., relay 1) transmits. Thus, the source is able to generate $(q+1)^{n-n_1}$ different sequences $\mathbf{x}_0$. The remaining n_1 slots are filled with off symbols N.[1]

Next, the maximum size of $\mathcal{W}_0, \mathcal{W}_1, \ldots, \mathcal{W}_{m-1}$ is derived. We immediately have

$$|\mathcal{W}_0| \leq (q+1)^{n-n_1}. \tag{2.4}$$

Both the source and the relays choose their messages uniformly and independently of each other. Hence, relay v needs to reserve $\prod_{i=0}^{v-1} |\mathcal{W}_i|$ sequences per transmission pattern of relay $v+1$ to represent any arriving messages $\{w_{0;b-v}, \ldots, w_{v-1;b-1}\}$. Arriving messages are encoded by each relay v with transmission patterns and a fixed number $k_v \in \{0, \ldots, n_v\}$ of transmission symbols. Observe that transmission patterns can only be used for encoding arriving messages. If relay v would encode own messages $w_{v;b}$ by means of transmission patterns, node $v-1$ would not know when node v listens since $w_{v;b}$ and, therefore, the transmission pattern of node v is not known by node $v-1$.

The remaining $n_v - k_v$ transmission symbols per transmission pattern are used by relay v for encoding own messages $w_{v;b}$. Hence

$$\prod_{i=0}^{v-1} |\mathcal{W}_i| \leq q^{k_v} \binom{n - n_{v+1}}{n_v} \quad \text{for all } v \in \mathcal{V} \setminus \{0, m\} \tag{2.5}$$

and

$$|\mathcal{W}_v| \leq q^{n_v - k_v}. \tag{2.6}$$

Example 2.2. We construct a code for a relay cascade composed of four nodes $\mathcal{V} = \{0, \ldots, 3\}$ such that $R_0 > 0$, $R_1 = 0$ and $R_2 > 0$. The transmission alphabet is binary, i.e., $q = 2$, and the selected code parameters are $n = 4$, $n_1 = 1$, $n_2 = 2$, $k_1 = 1$, $k_2 = 0$, $(n_3 = 0)$. Plugging these values in (2.4) to (2.6) yields $|\mathcal{W}_0| = |\mathcal{W}_2| = 4$ and $|\mathcal{W}_1| = 1$. Table 2.2a depicts possible codebooks $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2$ for nodes 0, 1, 2. Table 2.2b shows how to use the codebooks in order to send the messages given in column two and three.

w_0	$\mathcal{C}_0$			$\mathcal{C}_1$				$\mathcal{C}_2$
0	N0NN e	0NNN f	0NNN g	0NNN (a,e)	N0NN (b,f)	0NNN (c,e)	N0NN (d,f)	NBNC a
1	N1NN e	1NNN f	1NNN g	1NNN (a,e)	N1NN (b,f)	1NNN (c,e)	N1NN (d,f)	BNCN b
2	NN0N e	NN0N f	N0NN g	NN0N (a,g)	NNN0 (b,g)	NNN0 (c,g)	NN0N (d,g)	NBCN c
3	NN1N e	NN1N f	N1NN g	NN1N (a,g)	NNN1 (b,g)	NNN1 (c,g)	NN1N (d,g)	BNNC d

(a) Codebooks for nodes 0, 1, 2.

block b	$w_{0;b}$	$w_{2;b}$	$\mathbf{x}_0(b)$	$\mathbf{x}_1(b)$	$\mathbf{x}_2(b)$	$\hat{w}_{0;b}$	$\hat{w}_{2;b}$
$b=1$	3	-	NN1N	NNNN	NNNN	-	-
$b=2$	1	-	1NNN	NN1N	NNNN	-	-
$b=3$	2	0	NN0N	N1NN	0NN0	3	0
$b=4$	-	2	NNNN	NNN0	1N0N	1	2
$b=5$	-	3	NNNN	NNNN	N11N	2	3

(b) Illustration how to use the codebooks.

Table 2.2.: Illustration of the Timing Code constructed in Example 2.2.

Consider first $\mathcal{C}_2$. The four underlying transmission patterns, arbitrarily chosen from the $\binom{4}{2}$ possible patterns, are shown in the last column of Table 2.2a. Each transmission pattern is identified with a unique color $r \in \{a,b,c,d\}$. The $n_2 = 2$ binary transmission slots within each pattern are marked with $\mathsf{B},\mathsf{C} \in \{0,1\}$. The transmission patterns are used by node 2 to represent source messages w_0. For instance, pattern a represents $w_0 = 0$, pattern b represents $w_0 = 1$ and so forth. Own messages w_2 are encoded using the transmission symbols B and C according to the following assignment $w_2 \mapsto (\mathsf{B},\mathsf{C})$: $0 \mapsto (0,0)$, $1 \mapsto (0,1)$, $2 \mapsto (1,0)$, $3 \mapsto (1,1)$. Next, $\mathcal{C}_1$ is considered. Observe that $\mathcal{C}_1$ needs to be constructed such that node 1 is able to represent one random message from the 4-ary set $\mathcal{W}_0$ in each block independently from the transmission pattern used by node 2 in the same block. Hence, four codewords per transmission pattern a, b, c, d have to be constructed. Consider pattern a for instance. When node 2 uses pattern a, node 1 can encode its information in slots one and three. The following mapping $w_0 \mapsto (x_{1,1}, x_{1,3})$ can be chosen by node 1 to encode information in slot one and three: $0 \mapsto (0,\mathsf{N})$, $1 \mapsto (1,\mathsf{N})$, $2 \mapsto (\mathsf{N},0)$, $3 \mapsto (\mathsf{N},1)$. By allocating each of the four values of $(x_{1,1}, x_{1,3})$ to the N-slots of pattern a and by forcing node 1 to be off when node 2 transmits (i.e., allocating symbol N to slots 2 and 4), we obtain the codewords in the first column of $\mathcal{C}_1$. Applying the same procedure to pattern b, c, d yields column two, three, four of $\mathcal{C}_1$. The labels $(r,s) \in \{a,b,c,d\} \times \{e,f,g\}$ next to each codeword in $\mathcal{C}_1$ have the

following meaning. The first color r indicates the transmission pattern in $\mathcal{C}_2$ from which the codeword was constructed. The second color s refers to transmission patterns in $\mathcal{C}_1$ from which transmission patterns in $\mathcal{C}_0$ with the same color are constructed.

Finally, $\mathcal{C}_0$ is considered. Clearly, source node 0 has three time slots t_1, t_2, t_3 available per transmission block for encoding. Let $x_{0,t_1}, x_{0,t_2}, x_{0,t_3} \in \{0, 1, \mathsf{N}\}$ denote the symbols used by node 0 for encoding a particular message $w_0 \in \mathcal{W}_0$. The following mapping $w_0 \mapsto (x_{0,t_1}, x_{0,t_2}, x_{0,t_3})$ can be chosen for encoding: $0 \mapsto (0, \mathsf{N}, \mathsf{N})$, $1 \mapsto (1, \mathsf{N}, \mathsf{N})$, $2 \mapsto (\mathsf{N}, 0, \mathsf{N})$, $3 \mapsto (\mathsf{N}, 1, \mathsf{N})$. By allocating all values of $(x_{0,t_1}, x_{0,t_2}, x_{0,t_3})$ to the listen slots of codewords in $\mathcal{C}_1$ with second color $s \in \{e, f, g\}$ and by requiring that node 0 is off when node 1 transmits, we obtain the codewords in $\mathcal{C}_0$ colored with s.

It should be noted that merely four from 27 possible sequences are used in the mapping $w_0 \mapsto (x_{0,t_1}, x_{0,t_2}, x_{0,t_3})$. Hence, $\mathcal{C}_0$ could be designed such that node 0 is able to send $\lfloor 27/4 \rfloor$ additional messages to a sink at node 1, which corresponds to an additional rate of 0.6462 b/u.

Adjacent nodes are able to cooperate since each node knows the message(s) to be forwarded by the next node as well as the coding strategy applied by the next node. Hence, a node is aware of the codeword used by the next node and it can pick its codeword from the appropriate column. The codewords in block b are chosen as follows. Based on message $w_{0;b-2}$, the encoder at node 0 determines the color of $\mathbf{x}_2(b)$. Thus, it knows the first color r, of codeword $\mathbf{x}_1(b)$. Based on this information, the encoder at node 0 determines the second color s of $\mathbf{x}_1(b)$ using $w_{0;b-1}$. This color tells node 0 from which column in $\mathcal{C}_0$ it has to pick $\mathbf{x}_0(b)$, namely from a column whose codewords are colored with s. The precise choice within that column depends on the new source message $w_{0;b}$. Similarly, based on message $w_{0;b-2}$, the encoder at node 1 determines color r of $\mathbf{x}_2(b)$. Hence, it knows that $\mathbf{x}_1(b)$ needs to be chosen from a column of $\mathcal{C}_1$ where the entries have r as their first color. The precise choice within the column depends on message $w_{0;b-1}$. The encoder at node 2 knows $\{w_{0;b-2}, w_{2;b}\}$ at the beginning of block b. Message $w_{0;b-2}$ determines the transmission pattern of $\mathbf{x}_2(b)$ while $w_{2;b}$ specifies the transmission symbols B and C within the pattern.

We conclude the example by demonstrating how the codebooks $\mathcal{C}_0$, $\mathcal{C}_1$, $\mathcal{C}_2$ have to be used such that source node 0 is able to transmit messages $3, 1, 2$ to node 3 (and of course to relay nodes 1 and 2) while relay source 2 transmits messages $0, 2, 3$ to node 3.

We point out that the transmission strategy uses the agreement that each node picks its initial codeword from the first column of its codebook. The result is shown in Table 2.2b.

2.4.3. Achievable Rate Region

We now determine the rate region $\mathcal{R}$ due to the coding strategy proposed in Section 2.4.2. As usual, $R_i \stackrel{def}{=} \log\left(|\mathcal{W}_i|\right)/n$ where $0 \leq i < m$. Without loss of generality we assume that $R_0 > 0$ since the rate region of a cascade with $R_0 = 0$ is equal to the rate region of a shortened cascade where the first node with a positive rate becomes node 0 while all previous nodes are ignored. Hence, $0 < n_i < n$ for all $0 \leq i \leq m-1$. Moreover, $n_m = 0$ by assumption and $n_i \leq n - n_{i+1}$ because the number of transmission symbols used by node i cannot exceed the number of listening slots provided by node $i+1$. Using the abbreviations $p_i \stackrel{def}{=} n^{-1}(n - n_i)$ and $\bar{p}_i \stackrel{def}{=} 1 - p_i$ for the portion of time relay i listens and transmits, we obtain

$$0 < p_i < 1 \tag{2.7}$$

$$1 \leq p_i + p_{i+1} \tag{2.8}$$

$$p_m = 1 \tag{2.9}$$

for all $0 \leq i < m$. The set of probability vectors $(p_0, \ldots, p_m) \in \mathbb{R}^{m+1}$ characterized by (2.7) to (2.9) is denoted as $\mathcal{P}^{\star}_{m+1}$, which is a subset of the $m+1$-dimensional probability simplex $\mathcal{P}_{m+1}$. By the strong law of large numbers (or by noting that the portion of time node i listens is identical in each transmission block), we can identify the empirical probability p_i with $P_{X_i}(\mathsf{N})$ and $\bar{p}_i$ with $\cup_{q \in \mathcal{Q}} P_{X_i}(q)$. As a consequence of (2.8) and (2.9), all distributions in $\mathcal{P}^{\star}_{m+1}$ factorize as $P_{X_0} P_{X_1|X_0} \ldots P_{X_{m-1}|X_m} P_{X_m}$. Let $H(p_v)$ denote the binary entropy function evaluated at p_v. Using [29, Th. 1.4.5]

$$n^{-1} \log \binom{n}{n_v} = H(p_v) + o(1) \quad \text{for } n \to \infty, \tag{2.10}$$

it follows from (2.4) to (2.6) that

$$R_0 \leq p_1 \log(q+1) \tag{2.11}$$

$$\sum_{i=0}^{v} R_i \leq \bar{p}_v \log q + p_{v+1} H\left(\bar{p}_v p_{v+1}^{-1}\right) + o(1) \tag{2.12}$$

$$R_v \leq \bar{p}_v \log q \tag{2.13}$$

as $n \to \infty$. As an aside, (2.12) results from adding the logarithm of (2.5) to the logarithm of (2.6), dividing the result by n and applying (2.10). Inequality (2.12) is well-defined since $p_{v+1} \neq 0$ and $\bar{p}_v p_{v+1}^{-1} \in (0,1]$ due to (2.7) to (2.9). As $n \to \infty$, the achievable rate region $\mathcal{R}$ is given by

$$\mathcal{R} = Co\left(\bigcup_{\mathbf{p} \in \mathcal{P}_{m+1}^{\star}} \mathcal{R}_{\mathbf{p}}\right), \tag{2.14}$$

where $\mathcal{R}_{\mathbf{p}}$ indicates the region due to (2.11) to (2.13) for a particular $\mathbf{p} \in \mathcal{P}_{m+1}^{\star}$. The convex hull $Co(\cdot)$ takes time-sharing between different regions $\mathcal{R}_{\mathbf{p}}$ into account.

2.5. Capacity of Relay Cascades with One Source

In this section, we investigate the optimality of the coding strategy. We use the following notation. The complement of a set S within an ambient set is denoted as S^c, the power set of a set S is denoted as $\wp(S)$ and $X_S \overset{def}{=} \{X_i : i \in S\}$. Further, *probability mass function* is abbreviated as pmf.

2.5.1. A Simplified Cut-Set Bound for Relay Cascades

A well-known result, which bounds the rate of information flow from nodes in S^c to nodes in S is the so-called *cut-set bound*.

Lemma 2.1. [30, Chapter 14] Consider a general multiterminal network composed of $m + 1$ nodes and channel $P_{Y_0 \ldots Y_m | X_0 \ldots X_m}$. R_{ij} denotes the transmission rate between two nodes i and j. If the information rate R_{ij} is achievable, there is some joint probability distribution $P_{X_0 \ldots X_m}$ such that

$$\sum_{i \in \mathcal{S}^c, j \in \mathcal{S}} R_{ij} \leq I\left(X_{\mathcal{S}^c}; Y_{\mathcal{S}} | X_{\mathcal{S}}\right)$$

for all $\mathcal{S} \subset \{0, \ldots, m\}$.

Lemma 2.1 is now applied to the special case of a noise-free relay cascade. The following proof is of greater detail than originally presented in [2].

Lemma 2.2. [2, Chapter 5] Consider a noise-free relay cascade as described in Section 2.3. If the information rate R_v is achievable, there is some joint probability distribution $P_{X_v \ldots X_m}$ such that

$$\sum_{k=0}^{v} R_k \leq \max_{P_{X_v \ldots X_m}} \min_{v+1 \leq i \leq m} H(Y_i | X_i)$$

for all $v \in \mathcal{V} \setminus \{m\}$.

Proof. We show that it suffices to consider a subset of all network cuts. Observe that $(X_0, \ldots, X_{v-1}) - X_v - Y_m$ holds because of (2.2) and (2.3). Hence, an upper bound on the sum rate $\sum_{k=0}^{v} R_k$ due to Lemma 2.1 is given by

$$\sum_{k=0}^{v} R_k \leq \max_{P_{X_v \ldots X_m}} \min_{\mathcal{S} \in \mathcal{M}} I(X_v X_{\mathcal{S}^c}; Y_{\mathcal{S}} Y_m | X_{\mathcal{S}} X_m),$$

where $\mathcal{M} = \wp(\{v+1, \ldots, m-1\})$ and $\mathcal{S}^c$ is the complement of $\mathcal{S}$ in $\{v+1, \ldots, m-1\}$. Since the relay cascade is deterministic, we have

$$I(X_v X_{\mathcal{S}^c}; Y_{\mathcal{S}} Y_m | X_{\mathcal{S}} X_m) = H(Y_{\mathcal{S}} Y_m | X_{\mathcal{S}} X_m). \tag{2.15}$$

Now suppose that $\mathcal{S}$ is nonempty and let i denote the smallest integer in $\mathcal{S}$. By the chain rule for entropy

$$H(Y_{\mathcal{S}} Y_m | X_{\mathcal{S}} X_m) = H(Y_i | X_{\mathcal{S}} X_m) + H(Y_{\mathcal{S} \setminus \{i\}} | X_{\mathcal{S}} X_m Y_i) + H(Y_m | X_{\mathcal{S}} X_m Y_{\mathcal{S}}) \tag{2.16}$$

$$\geq H(Y_i | X_{\mathcal{S}} X_m), \tag{2.17}$$

where (2.17) follows from the non-negativity of entropy. For each cut $\mathcal{S}$ with smallest entry i, the cut $\mathcal{S}_i \stackrel{def}{=} \{i, \ldots, m-1\}$ yields

$$H(Y_{\mathcal{S}_i} Y_m | X_{\mathcal{S}_i} X_m) \leq H(Y_{\mathcal{S}} Y_m | X_{\mathcal{S}} X_m). \tag{2.18}$$

Inequality (2.18) is true because the second and third term on the right hand side of

(2.16) drop out (since $\mathcal{S}$ is replaced by $\mathcal{S}_i$ and channel model (2.2 holds). Moreover, the remaining term $H(Y_i|X_{\mathcal{S}_i}X_m)$ satisfies $H(Y_i|X_{\mathcal{S}}X_m) \geq H(Y_i|X_{\mathcal{S}_i}X_m)$ since $\mathcal{S} \subseteq \mathcal{S}_i$ and conditioning does not increase entropy. It remains to consider $\mathcal{S} = \emptyset$ in (2.15), which yields $H(Y_m|X_m)$. Putting everything together, we see that $\sum_{k=0}^{v} R_k$ is upper bounded by

$$\sum_{k=0}^{v} R_k \leq \max_{P_{X_v \ldots X_m}} \min_{v+1 \leq i \leq m} H(Y_i|X_{\mathcal{S}_i}X_m)$$

$$\leq \max_{P_{X_v \ldots X_m}} \min_{v+1 \leq i \leq m} H(Y_i|X_i).$$

The last inequality follows again from the property that conditioning does not increase entropy. ∎

2.5.2. Main Results

Theorem 2.1 was derived in [2] for the special cases $q = 2, 3$. We generalize the result in the following and present a rigorous proof. This result also appeared in [1] in the sense that the (general) cut-set bound was shown to be the capacity. Theorem 2.1 refines the result in the sense that it shows that only m particular cuts suffice. Moreover, it uses a new constructive achievability scheme and gives a more explicit characterization (2.20).

Theorem 2.1. The capacity of a noise-free relay cascade as described in Section 2.3 with a single source-destination pair (namely node 0 and node m) and $m-1$ half-duplex-constrained relays is given by

$$C_{m-1}(q) = \max_{P_{X_0 \ldots X_m}} \min_{1 \leq i \leq m} H(Y_i|X_i), \tag{2.19}$$

where the maximization is over all $P_{X_0 \ldots X_m}$ which have marginal pmfs $P_{X_{i-1}X_i}$ as shown in Table 2.3a and 2.3b. Under consideration of the optimal input distribution as stated in Table 2.3a and 2.3b, (2.19) becomes

$$C_{m-1}(q) = \max_{p_1, \ldots, p_{m-1}} \min\left\{ p_1 \log(q+1), \min_{1 \leq i \leq m-1} \left\{ \bar{p}_i \log q + p_{i+1} H\left(\bar{p}_i p_{i+1}^{-1}\right) \right\} \right\}, \tag{2.20}$$

where $0 < p_i < 1$, $p_m = 1$ and $p_i + p_{i+1} \geq 1$ for all $i \in \{1, \ldots, m-1\}$.

X_{i-1} \ X_i	0	$\cdots$	$q-1$	N
0	0	$\cdots$	0	$\bar{p}_{i-1}/q$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$q-1$	0	$\cdots$	0	$\bar{p}_{i-1}/q$
N	$\bar{p}_i/q$	$\cdots$	$\bar{p}_i/q$	$p_i - \bar{p}_{i-1}$

(a) The Marginal Pmfs $P_{X_0 X_1}, \ldots, P_{X_{m-1} X_m}$.

X_0 \ X_1	0	$\cdots$	$q-1$	N
0	0	$\cdots$	0	$p_1/(q+1)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	
$q-1$	0	$\cdots$	0	$\vdots$
N	$\bar{p}_1/q$	$\cdots$	$\bar{p}_1/q$	$p_1/(q+1)$

(b) Pmf $P_{X_0 X_1}$ from part (a) with $p_0 = (1 + q - qp_1)(1+q)^{-1}$.

Table 2.3.: The Marginal Pmfs $P_{X_0 X_1}, \ldots, P_{X_{m-1} X_m}$ of the Input Pmf.

Proof. By Lemma 2.2, we have

$$C_{m-1}(q) \leq \max_{P_{X_0 \cdots X_m}} \min_{1 \leq i \leq m} H(Y_i | X_i). \tag{2.21}$$

In the maximization (2.21), we only need to consider input pmfs $P_{X_0 \cdots X_m}$ with marginal pmfs $P_{X_0 X_1}, \ldots, P_{X_{m-1} X_m}$ as stated in Table 2.3a. The zero probability assignments in Table 2.3a result from the following property [31]: a channel input can be neglected if it produces the same channel output as another channel input and this with the same probabilities. Consider the first column of Table 2.3a. For all $k \in \mathcal{X}$, the inputs $(X_{i-1}, X_i) = (k, 0)$ yield $Y_i = 0$ with probability 1. Hence, we can assume a zero probability mass for all but one input $(X_{i-1}, X_i) = (k, 0)$. The same reasoning can be applied to the second till q^{th} column of Table 2.3a. In summary, it is without loss to keep only one positive entry in each of the first q columns of Table 2.3a. We decide to assign a positive probability mass to the last entry of each of the first q columns. Let us now address the last column of Table 2.3a. By symmetry, any permutation of the transmission

symbols $X_{i-1} \in \mathcal{Q}$ yields the same information flow between node $i-1$ and node i. Hence, we can choose $P_{X_{i-1}X_i}(k, \mathsf{N}) = P_{X_{i-1}X_i}(l, \mathsf{N})$ for all $k, l \in \mathcal{Q}$. Consequently, $P_{X_{i-1}X_i}(k, \mathsf{N}) = \bar{p}_{i-1}/q$ for all $k \in \mathcal{Q}$. The constraints $0 < p_i < 1$, $p_i + p_{i-1} \geq 1$, $p_m = 1$ guarantee that Table 2.3a is a proper probability mass functions and that node m is always quiet. The assumption $p_i \neq \{0, 1\}$ for all $0 \leq i < m$ is without loss since otherwise node i cannot use the complete alphabet $\mathcal{X}$. It is now easy to check that

$$H(Y_{i+1}|X_{i+1}) = \bar{p}_i \log q + p_{i+1} H\left(\bar{p}_i p_{i+1}^{-1}\right) \quad \text{for all } 0 \leq i \leq m-1. \tag{2.22}$$

Observe that p_0 occurs only in $H(Y_1|X_1)$. Moreover, $H(Y_1|X_1)$ is concave in p_0. Hence, $H(Y_1|X_1)$ can be maximized with respect to p_0 without affecting any of the other $H(Y_{i+1}|X_{i+1})$. Setting the partial derivative

$$\frac{\partial H(Y_1|X_1)}{\partial p_0} = -\log q + \log\left(\frac{1 - p_0}{p_0 + p_1 - 1}\right)$$

equal to zero yields $p_0 = (1 + q - qp_1)(1 + q)^{-1}$ and the marginal pmf $P_{X_0 X_1}$ becomes as shown in Table 2.3b. Moreover, $H(Y_1|X_1)$ becomes

$$H(Y_1|X_1) = p_1 \log(q + 1). \tag{2.23}$$

By replacing the conditional entropy functions in (2.21) with (2.22) and (2.23), we obtain

$$
\begin{aligned}
C_{m-1}(q) &\leq \max_{P_{X_0 \dots X_m}} \min_{1 \leq i \leq m} H(Y_i|X_i) \\
&= \max_{p_1, \dots, p_{m-1}} \min\left\{ p_1 \log(q+1), \min_{1 \leq i \leq m-1}\left\{ \bar{p}_i \log q + p_{i+1} H\left(\bar{p}_i p_{i+1}^{-1}\right) \right\} \right\},
\end{aligned}
\tag{2.24}
$$

where $0 < p_i < 1$, $p_m = 1$ and $p_i + p_{i+1} \geq 1$ for all $i \in \{1, \dots, m-1\}$. Recall that $R_i = 0$ for all $i \in \mathcal{V} \setminus \{0\}$. Using (2.11) and (2.12), we have

$$C_{m-1}(q) \geq \max_{p_1, \dots, p_{m-1}} \min\left\{ p_1 \log(q+1), \min_{1 \leq i \leq m-1}\left\{ \bar{p}_i \log q + p_{i+1} H\left(\bar{p}_i p_{i+1}^{-1}\right) \right\} \right\}, \tag{2.25}$$

where the maximization is with respect to $\mathcal{P}_{m+1}^{\star}$. Note that $\mathcal{P}_{m+1}^{\star}$ includes the set of pmfs defined by Table 2.3a and 2.3b. In summary, (2.24) and (2.25) imply (2.19)

and (2.20).   ■

Remark 2.1.   (i) A more intuitive explanation for the zero probability assignments in
Table 2.3a and 2.3b is the following. Assume node i is transmitting, i.e., $X_i \in \mathcal{Q}$.
According to the underlying channel model, node i is not able to listen to the
input of node $i-1$ and, consequently, node $i-1$ should not transmit when node i
transmits.

 (ii) The marginal pmf $P_{X_0 X_1}$ as shown in Fig. 2.3b demonstrates that in order to
achieve the maximum information flow from source node 0 to relay 1, the source
has to encode with uniformly distributed input symbols when relay 1 listens, i.e.,
$P_{X_0|X_1=\mathsf{N}}$ is the uniform distribution. One could ask why $P_{X_{i-1}|X_i=\mathsf{N}}$ is not nec-
essarily uniform for $i > 1$. However, in contrast to the source node, relay $i-1$
receives information. The amount of received information depends on the fraction
of listening time provided by relay $i-1$. Thus, choosing uniformly distributed
inputs $(X_{i-1}, X_i) = (k, \mathsf{N})$, $k \in \mathcal{X}$, maximizes the rate on link $(i-1, i)$ but might
reduce the rate on link $(i-2, i-1)$.

(iii) Capacity expression (2.19) in Theorem 2.1 could also have been obtained by ap-
plying the DF-rate of Xie and Kumar [28] to the network model considered here.
However, we show achievability by a constructive coding argument while Xie and
Kumar use a pure random coding argument in their proof.

Theorem 2.2. For $m \to \infty$, i.e., for an unbounded number of relays, the capacity of a
noise-free and half-duplex-constrained relay cascade as described in Section 2.3 with a
single source-destination pair is given by

$$C_\infty(q) = \log_2 \left(\frac{1 + \sqrt{4q+1}}{2} \right) \text{ b/u.} \tag{2.26}$$

For the proof of Theorem 2.2, we need the following lemma.

Lemma 2.3. Consider a noise-free relay cascade of finite length as described in Sec-
tion 2.3 with a single source-destination pair (namely node 0 and node m) and $m-1$

half-duplex-constrained relays. There exists a capacity achieving input pmf $P_{X_0 \dots X_m}$ such that $C_{m-1}(q) = H(X_{m-1})$.

Proof. Recall the capacity expression stated in Theorem 2.1 and assume $H(Y_m|X_m) > C_{m-1}(q)$. We will show that $P_{X_0 \dots X_m}$ can be changed such that $H(Y_m|X_m)$ equals $C_{m-1}(q)$ without forcing any of the $H(Y_i|X_i)$, $1 \leq i \leq m-1$, to decrease. The optimal input pmf as stated in Table 2.3a and 2.3b is assumed in the following. Hence, $H(Y_m|X_m) = H(X_{m-1}|X_m) = H(X_{m-1})$ and $H(Y_i|X_i) = H(X_{i-1}|X_i)$ for all $1 \leq i \leq m-1$. Recall (2.22) and (2.23), i.e., $H(X_0|X_1) = p_1 \log(q+1)$ and $H(X_{i-1}|X_i) = \bar{p}_{i-1} \log q + p_i H\left(\bar{p}_{i-1} p_i^{-1}\right)$, $2 \leq i \leq m$ and $p_m = 1$. The assertion is clear for $m = 2$. In this case, the capacity is simply the intersection of $H(X_0|X_1) = p_1 \log(q+1)$ and $H(X_1) = \bar{p}_1 \log q + H(\bar{p}_1)$. In particular, $C_1(q) = H(X_1)$. Let $m > 2$. Changing $H(X_{m-1})$ by means of varying p_{m-1} does not affect $H(X_{i-1}|X_i)$, $1 \leq i \leq m-2$, since both expressions depend on different variables. Therefore, it is enough to consider $H(X_{m-2}|X_{m-1})$. The maximum of $H(X_{m-1})$ is attained at $p_{m-1} = 1/(q+1)$. Further, $H(X_{m-1})$ is continuously decreasing to zero when p_{m-1} runs through the interval $[1/(q+1), 1]$. Assume that $p_{m-1} \geq 1/(q+1)$ and that $H(X_{m-1}) > C_{m-1}(q)$. In order to decrease $H(X_{m-1})$ to $C_{m-1}(q)$, we just have to increase p_{m-1} appropriately. Observe that increasing p_{m-1} does not decrease $H(X_{m-2}|X_{m-1})$ since its partial derivative with respect to p_{m-1} is

$$\frac{\partial H(X_{m-2}|X_{m-1})}{\partial p_{m-1}} = \log\left(\frac{p_{m-1}}{p_{m-1} + p_{m-2} - 1}\right), \tag{2.27}$$

which is always non-negative. This follows from $0 \leq 1 - p_{m-2} \leq p_{m-1}$, where the right inequality holds since $P_{X_{m-2} X_{m-1}}(\mathsf{N}, \mathsf{N}) = p_{m-1} - \bar{p}_{m-2} \geq 0$. Hence, the argument of the logarithm in (2.27) is always greater or equal than one. Now consider the case $p_{m-1} < 1/(q+1)$ and $H(X_{m-1}) > C_{m-1}(q)$. This case can be excluded. To decrease $H(X_{m-1})$ to $C_{m-1}(q)$, we can increase p_{m-1} to the appropriate value in the interval $[1/(q+1), 1]$. By the non-negativity of (2.27), $H(X_{m-2}|X_{m-1})$ can only increase. ∎

Proof of Theorem 2.2. We first show that the series of capacities $(C_m(q))_{m \in \mathbb{N}}$ is monotonically decreasing and bounded, hence convergent. Let $k > l$ be two positive integers. By Lemma 2.3, there exist two input pmfs such that $C_k(q) = H(X_k)$ and $C_l(q) = H(X_l)$.

Then

$$0 \leq C_k(q) \leq H(X_l|X_{l+1}) \leq H(X_l) \leq C_l(q), \tag{2.28}$$

where both entropy functions are evaluated with respect to the $C_k(q)$ achieving input pmf. The second inequality (from left) follows from (2.19) and the third inequality from the property that conditioning does not increase entropy. The fourth inequality is due to the fact that the $C_k(q)$-achieving input pmf $P_{X_0 \ldots X_k}$ (if marginalized to $P_{X_0 \ldots X_l}$) does not necessarily achieve $C_l(q)$. This shows that $(C_m(q))_{m \in \mathbb{N}}$ is monotonically decreasing. Boundedness holds since $0 \leq C_m(q) \leq C_1(q) < \infty$ for all $m \in \mathbb{N}$. Thus, $C_\infty(q) \stackrel{def}{=} \lim_{m \to \infty} C_m(q)$ exists. Then for any $\epsilon > 0$ there exists a $N(\epsilon) \in \mathbb{N}$ such that

$$|C_\infty(q) - C_l(q)| < \epsilon, \quad \forall l > N(\epsilon). \tag{2.29}$$

Hence

$$|H(X_{l+1}|X_{l+2}) - H(X_l|X_{l+1})|$$
$$\leq |C_\infty(q) - H(X_{l+1}|X_{l+2})| + |C_\infty(q) - H(X_l|X_{l+1})| \tag{2.30}$$
$$< 2\epsilon \tag{2.31}$$

for all $l > N(\epsilon)$, where (2.30) follows from the triangle inequality while (2.31) follows from (2.28) and (2.29). Consequently, the $C_\infty(q)$-achieving input pmf satisfies

$$H(X_{l+1}|X_{l+2}) - H(X_l|X_{l+1}) = 0 \quad \text{as } l \to \infty. \tag{2.32}$$

By entirely the same arguments as given in the proof of Theorem 2.1, we can assume that the marginals $P_{X_l X_{l+1}}$ and $P_{X_{l+1} X_{l+2}}$ of the $C_\infty(q)$ achieving pmf are of the form outlined in Table 2.3a. Then (2.32) is satisfied if and only if $p_l = p_{l+1} = p_{l+2} \stackrel{def}{=} p$. This choice results in marginal pmfs $P_{X_l X_{l+1}}$ as illustrated in Table 2.4 for all $l \in \mathbb{N}_0$. Thus, using (2.22), we have

$$H(X_l|X_{l+1}) = \bar{p} \log q + p H \left(\bar{p} p^{-1} \right) \quad \text{for all } l \in \mathbb{N}_0. \tag{2.33}$$

Hence, $C_\infty(q)$ is equal to the maximum of (2.33), which is (determining the zero of the

X_l \ X_{l+1}	0	$\cdots$	$q-1$	N
0	0	$\cdots$	0	$\bar{p}/q$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$q-1$	0	$\cdots$	0	$\bar{p}/q$
N	$\bar{p}/q$	$\cdots$	$\bar{p}/q$	$2p-1$

Table 2.4.: The Marginal Pmf $P_{X_l X_{l+1}}$ of the $C_\infty(q)$-Achieving Input Pmf, $l \in \mathbb{N}_0$.

first derivative of (2.33) and plugging it into (2.33))

$$C_\infty(q) = \log\left(\frac{1 + \sqrt{4q+1}}{2}\right)$$

attained at

$$p = \frac{1}{2}\left(1 + \frac{1}{\sqrt{4q+1}}\right).$$

∎

Remark 2.2. (i) $C_\infty(1) = 0.6942$ b/u is equal to the logarithm of the *golden ratio* and $C_\infty(2)$ is 1 b/u.

(ii) The term *time-sharing* denotes a transmission strategy where nodes send and receive half of the time, organized in deterministic fashion. The maximum achievable rate with time-sharing is given by $R_{ts}(q) \stackrel{def}{=} 0.5\log(q+1)$ b/u. For $q = 1, 2$ we have $R_{ts}(1) = 0.5$ b/u and $R_{ts}(2) \approx 0.7925$ b/u. Clearly, $C_\infty(q)$ is a lower bound on the capacity of any finite length cascade where the transmission alphabet has size q. A comparison of $R_{ts}(1)$ and $R_{ts}(2)$ with $C_\infty(1)$ and $C_\infty(2)$ shows that pre-determined time-sharing falls considerably short of the capacity for small transmission alphabets. For very large transmission alphabets, we have $\lim_{q\to\infty}[C_\infty(q) - R_{ts}(q)] = 0$. That is, the gap between the rates due to time-sharing and timing becomes negligible.

(iii) The stochastic process $\{X_l : l \geq 0\}$ under the pmf depicted in Table 2.4 is a stationary, irreducible Markov chain with finite state space $\mathcal{X}$. Moreover, $C_\infty(q)$ is equal to the *entropy rate* [30, Chapter 4] of $\{X_l : l \geq 0\}$.
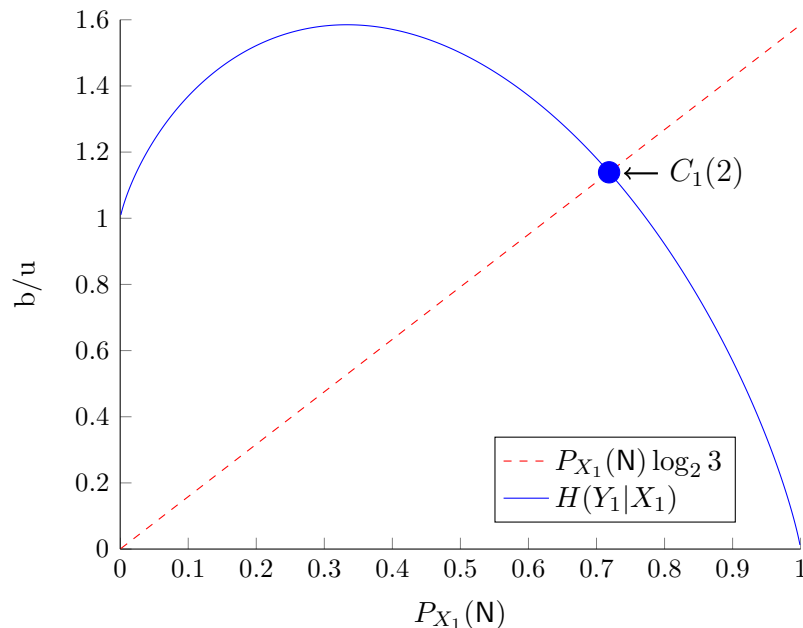
Figure 2.3.: Graphical solution of optimization problem (2.34).

We now provide numerical capacity results for various scenarios by means of Theorem 2.1. In particular, we show how to obtain the capacity of a half-duplex-constrained relay cascade with one source-destination pair for an arbitrary number of relays.

Example 2.3. Let us first consider a relay cascade with $\mathcal{V} = \{0, 1, 2\}$, $q = 2$ and $R_1 = 0$, i.e., source node 0 intends to communicate with sink node 2 via a half-duplex-constrained relay. By Theorem 2.1 and the optimum input pmf stated in Table 2.3b, we have

$$C_1(2) = \max_{P_{X_0 X_1 X_2}} \min \left\{ P_{X_1}(\mathsf{N}) \log_2 3, H(X_1) \right\}. \tag{2.34}$$

Problem (2.34) exhibits a single degree of freedom and is readily solved by finding the intersection of the two functions $P_{X_1}(\mathsf{N}) \log 3$ and $H(X_1)$ (see Fig. 2.3). Hence, $C_1(2) = 1.1389$ b/u attained at $P_{X_1}(\mathsf{N}) = 0.7185$.

Remark 2.3.   (i) Assume the relay in Example 2.3 does not have the capability to detect when the source uses off symbols N. This behavior is modeled by setting $P_{X_0 X_1}(\mathsf{N}, \mathsf{N}) = 0$. An identical approach as taken in Example 2.3 shows that the capacity under the described constraint equals 0.8295 b/u. Observe that this

quantity is still larger than the corresponding time-sharing rate of $R_{ts}(2) \approx 0.7925$ bit per use.

(ii) For $q = 1$, the approach taken in Example 2.3 yields $C_1(1) = 0.7729$ b/u attained at $P_{X_1}(\mathsf{N}) = 0.7729$. The capacity of this specific model has also been obtained in [31]. Therein, the focus was not on half-duplex-constrained transmission but on finding the capacity of certain classes of deterministic relay channels. In [1], the same channel model was considered and the author also noticed that the capacity equals 0.7729 b/u. A simple coding scheme was outlined which approaches $\frac{2}{3}$ b/u, and extensions using Huffman or arithmetic source coding are claimed.

Example 2.4. In order to compute $C_{m-1}(q)$ for $m > 2$, we transform (2.19) into a convex program with linear cost function $H(Y_1|X_1)$ and convex inequality constraints $H(Y_1|X_1) - H(Y_{i+1}|X_{i+1}) \leq 0$ for all $i \in \{1, \ldots, m-1\}$. The resulting program reads as

$$
\begin{aligned}
\text{maximize} \quad & p_1 \log_2(q+1) \\
\text{subject to} \quad & p_1 \log_2(q+1) - \bar{p}_i \log_2 q - p_{i+1} H\left(\bar{p}_i p_{i+1}^{-1}\right) \leq 0 \\
& 1 - \sum_{j=i}^{i+1} p_j \leq 0 \\
& p_i \in (0,1), \; p_m = 1.
\end{aligned}
$$

By adopting a standard algorithm for constrained optimization problems (e.g., MATLAB's *fmincon* – a gradient-based method), the capacity $C_{m-1}(q)$ was computed for various values of $m-1$ and $q = 1, 2$. A brief summary is provided in Table 2.5.

Table 2.5.: Capacity Results for Cascades with a Single Source.

$m-1$	1	2	3	10	40	100	∞
$C_{m-1}(1)$ [b/u]	0.7729	0.7324	0.7173	0.6981	0.6946	0.6943	0.6942
$C_{m-1}(2)$ [b/u]	1.1389	1.0665	1.0400	1.0066	1.0006	1.0001	1
$C_{m-1}(1)/R_{ts}(1)[\%]$	154.6	146.5	143.5	139.6	138.9	138.9	138.8
$C_{m-1}(2)/R_{ts}(2)[\%]$	143.7	134.6	131.2	127.0	126.3	126.2	126.2

We observe that $C_{m-1}(q)/R_{ts}(q)$, the relative advantage of timing codes over simple

time-sharing, decreases when $m - 1$ increases and saturates at 138.8% and 126.2% for $q = 1$ and $q = 2$, respectively.

2.6. Capacity and Rate Region of Relay Cascades with Multiple Sources

Based on the coding strategy introduced in Section 2.4.2, we derive an achievable rate region for cascades where every node except of the sink node can be a source. Let $\mathcal{C}^{\uparrow}_{m-1}$ denote the set of all rate vectors $\mathbf{R} \in \mathbb{R}^m_{\geq 0}$ satisfying

$$\mathcal{C}^{\uparrow}_{m-1} \stackrel{def}{=} \left\{ \mathbf{R} \in \mathbb{R}^m_{\geq 0} : \sum_{k=0}^{v} R_k \leq H(Y_{v+1}|X_{v+1}), \quad \text{for all } 0 \leq v < m \right\}. \tag{2.35}$$

By Lemma 2.2, $\mathcal{C}^{\uparrow}_{m-1}$ is the cut-set region of a line network composed of a source, a sink and $m - 1$ half-duplex-constrained relays where the first m nodes of the line can have a positive transmission rate. Without loss of generality, we assume input pmfs $P_{X_0 \ldots X_m}$ such that the marginal pmfs $P_{X_{v-1} X_v}$ are as shown in Table 2.3. Taking into account the resulting entropy functions (2.22) and (2.23), $\mathcal{C}^{\uparrow}_{m-1}$ reads as

$$\mathcal{C}^{\uparrow}_{m-1} = \left\{ \mathbf{R} \in \mathbb{R}^m_{\geq 0} : \left\{ \begin{array}{l} R_0 \leq p_1 \log(q+1) \\ \sum_{k=0}^{v} R_k \leq \bar{p}_v \log q + p_{v+1} H\left(\bar{p}_v p_{v+1}^{-1}\right), \quad \text{for all } 0 < v < m \end{array} \right. \right\}.$$

Let $\mathcal{C}^{\uparrow}_{m-1,s}$ denote the set of all rate vectors $\mathbf{R} \in \mathbb{R}^m_{\geq 0}$ satisfying

$$\mathcal{C}^{\uparrow}_{m-1,s} \stackrel{def}{=} \left\{ \mathbf{R} \in \mathbb{R}^m_{\geq 0} : \left\{ \begin{array}{l} R_v = 0, \quad \text{for all } 0 \leq v < s \\ \sum_{k=s}^{v} R_k \leq H(Y_{v+1}|X_{v+1}), \quad \text{for all } s \leq v < m \end{array} \right. \right\},$$

where $0 \leq s < m$. Observe that $\mathcal{C}^{\uparrow}_{m-1,s}$ is the cut-set region of a line network composed of $m + 1$ nodes where the first s nodes are constrained to have a transmission rate equal to 0. Similar to before, we can assume that $P_{X_s \ldots X_m}$ marginalizes according to Table 2.3.

In particular,

$$
\mathcal{C}_{m-1,s}^{\uparrow} = \left\{ \mathbf{R} \in \mathbb{R}_{\geq 0}^{m} : \begin{cases} R_v = 0, \quad \text{for all } 0 \leq v < s \\ R_s \leq p_{s+1} \log(q+1) \\ \sum_{k=s}^{v} R_k \leq \bar{p}_v \log q + p_{v+1} H\left(\bar{p}_v p_{v+1}^{-1}\right), \quad \text{for all } s \leq v < m \end{cases} \right\}.
$$

In contrast to the single source case, constraint (2.13) has to be considered. These constraints stem from the coding procedure, namely that all nodes with indices greater zero can only use transmission symbols for encoding own information. The usage of transmission patterns for encoding own information would render cooperation with the previous node impossible. These constraints are taken into account by introducing the sets

$$
\mathcal{K}_{m-1,s} \overset{def}{=} \left\{ \mathbf{R} \in \mathbb{R}_{\geq 0}^{m} : R_v \leq \bar{p}_v \log q, \quad \text{for all } s < v < m \right\}, \tag{2.36}
$$

where $0 \leq s < m$. Then

$$
\mathcal{R}_{m-1,s} \overset{def}{=} \bigcup_{P_{X_s \dots X_m}} \mathcal{C}_{m-1,s}^{\uparrow} \cap \mathcal{K}_{m-1,s} \tag{2.37}
$$

is a set of achievable rates of a line network composed of $m + 1$ nodes where the first s nodes are constrained to have a transmission rate equal to 0. In summary, we have shown the following theorem.

Theorem 2.3. Consider a noise-free relay cascade as described in Section 2.3 where each node except of the sink node can act as a source. The achievable rate region $\mathcal{R}_{m-1}$ due to the coding strategy outlined in Section 2.4.2 is given by

$$
\mathcal{R}_{m-1} = Co\left(\bigcup_{s=0}^{m-1} \mathcal{R}_{m-1,s}\right), \tag{2.38}
$$

where $\mathcal{R}_{m-1,s}$ is defined in (2.37). The union in (2.37) is over all $P_{X_s \dots X_m}$ with marginal pmfs $P_{X_{v-1} X_v}$, $s < v \leq m$, as depicted in Table 2.3.

As a final remark, observe that all boundary points of the cut-set region $\mathcal{C}_{m-1}^{\uparrow}$ that are achievable under the constraints (2.36) are capacity points.

Example 2.5. A half-duplex line network with node set $\mathcal{V} = \{0, 1, 2\}$ is considered (see
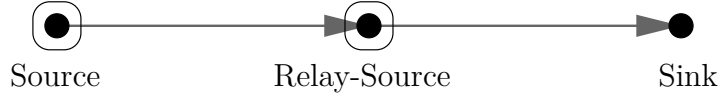
Figure 2.4.: A noiseless relay cascade composed of three nodes with two sources.

Fig. 2.4). Node 0 and node 1 are allowed to have a positive transmission rate. We first derive an explicit expression of the boundary of the cut-set region

$$\mathcal{C}_1^{\uparrow} = \left\{ \mathbf{R} \in \mathbb{R}_{\geq 0}^2 : \left\{ \begin{array}{l} R_0 \leq p_1 \log(q+1) \\ R_0 + R_1 \leq \bar{p}_1 \log q + H\left(p_1\right) \end{array} \right\} \right\}. \tag{2.39}$$

Observe that the maximum of the sum rate bound $\bar{p}_1 \log q + H\left(p_1\right)$ is $\log(q+1)$, attained at $p_1 = (q+1)^{-1}$. Hence, $0 \leq R_0 \leq \frac{1}{q+1} \log(q+1)$ and $0 \leq R_0 + R_1 \leq \log(q+1)$. Since every boundary point satisfies the defining inequalities of $\mathcal{C}_1^{\uparrow}$ with equality, we can express a part of the boundary as

$$R_1 = \log(q+1) - R_0 \quad \text{for all } 0 \leq R_0 \leq \frac{1}{q+1} \log(q+1). \tag{2.40}$$

The remaining part of the boundary of $\mathcal{C}_1^{\uparrow}$ is derived as follows. Applying the substitution $p_1 = R_0 / \log(q+1)$, which results from the first inequality in (2.39), to the second inequality in (2.39) yields

$$R_1 = \left(1 - \frac{R_0}{\log(q+1)}\right) \log q + H\left(\frac{R_0}{\log(q+1)}\right) - R_0 \quad \frac{1}{q+1} \log(q+1) < R_0 \leq C_1(q), \tag{2.41}$$

where $C_1(q)$ denotes the capacity of the considered line network if $R_1 = 0$.

We proceed with the computation of $\mathcal{R}_1 = Co\left(\mathcal{R}_{1,0} \cup \mathcal{R}_{1,1}\right)$. By (2.37), $\mathcal{R}_{1,0}$ satisfies

$$0 \leq R_0 \leq p_1 \log(q+1) \tag{2.42}$$

$$0 \leq R_0 + R_1 \leq \bar{p}_1 \log q + H(p_1) \tag{2.43}$$

$$0 \leq R_1 \leq \bar{p}_1 \log q. \tag{2.44}$$

Observe that (2.42) and (2.43) correspond to the cut-set region $\mathcal{C}_1^{\uparrow}$. Hence, all rate

boundary points of $\mathcal{R}_{1,0}$ are characterized by

$$R_1 = \left(1 - \frac{R_0}{\log(q+1)}\right)\log q + H\left(\frac{R_0}{\log(q+1)}\right) - R_0, \tag{2.45}$$

where $0 \leq R_1 \leq \bar{p}_1 \log q$. Using the same substitution as before, the latter constraint can be written as

$$0 \leq R_1 \leq \left(1 - \frac{R_0}{\log(q+1)}\right)\log q. \tag{2.46}$$

Equality on the right side of (2.46) implies, taking into account (2.45), that R_0 has to satisfy

$$R_0 = H\left(\frac{R_0}{\log(q+1)}\right). \tag{2.47}$$

Let $R_0^\star$ be the fixpoint of (2.47). Two cases can be distinguished: $R_0 < R_0^\star$ and $R_0 \geq R_0^\star$. In first case, the left hand side of (2.47) is strictly smaller than the right hand side due to the concavity of the binary entropy function. Therefore, the values for R_1 due to (2.45) violate (2.46). Hence, we can ignore the first case. In the second case, the left hand side of (2.47) is at least as large than the right hand side. Hence, (2.46) is satisfied and we keep $R_0 \geq R_0^\star$. It remains to consider $\mathcal{R}_{1,1}$, the set of all rate vectors satisfying $R_0 = 0$ and $0 \leq R_1 \leq \log(q+1)$. Clearly, $(R_0, R_1) = (0, \log(q+1))$ is the rate boundary of $\mathcal{R}_{1,1}$.

The rate boundary of $\mathcal{R}_1$ now follows by taking the convex hull of the rate boundary of $\mathcal{R}_{1,0}$ and $\mathcal{R}_{1,1}$, i.e, the convex hull of all points satisfying (2.45) for $R_0^\star \leq R_0 \leq C_1(q)$ and the point $(R_0, R_1) = (0, \log(q+1))$. Hence,

$$R_1 = \begin{cases} \left(\dfrac{\log q}{R_0^\star} - \dfrac{\log q}{\log(q+1)} - \dfrac{\log(q+1)}{R_0^\star}\right)R_0 + \log(q+1), & 0 \leq R_0 < R_0^\star \quad (2.48\text{a}) \\[2ex] \left(1 - \dfrac{R_0}{\log(q+1)}\right)\log q + H\left(\dfrac{R_0}{\log(q+1)}\right) - R_0, & R_0^\star \leq R_0 \leq C_1(q) \quad (2.48\text{b}) \end{cases}$$

where (2.48a) is just the straight line between the two rate vectors $(0, \log(q+1))$ and $(R_0, R_1) = \left(R_0^\star, \left(1 - \frac{R_0^\star}{\log(q+1)}\right)\log q\right)$.

The cut-set region $\mathcal{C}_1^\uparrow$ and the rate region $\mathcal{R}_1$ (in b/u) are depicted in Fig. 2.5 for a binary transmission alphabet $q = 2$. In this case, $R_0^\star$ equals 0.9654 b/u. Hence, the cut-set
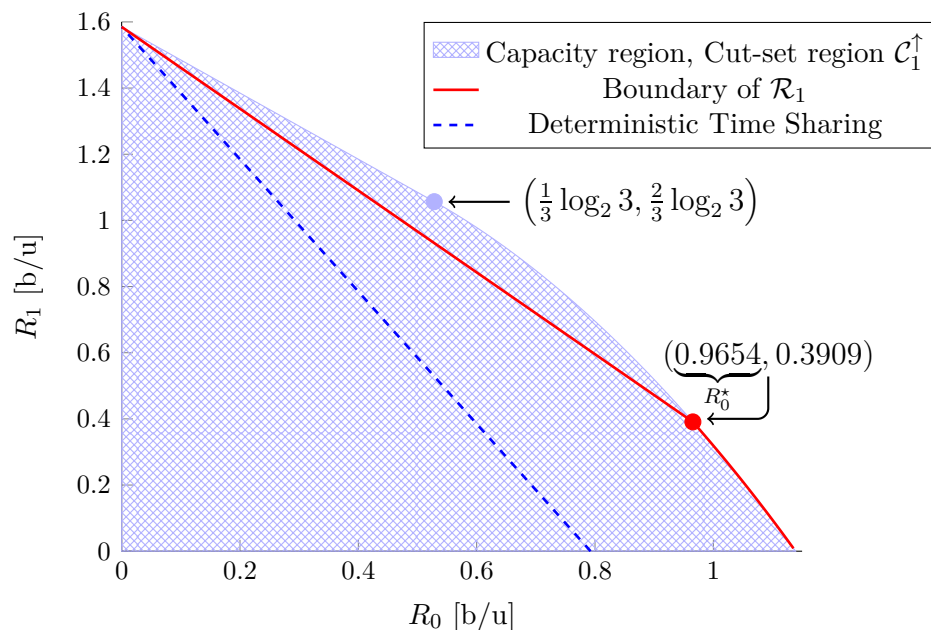
Figure 2.5.: The rate regions derived in Example 2.5.

bound is achievable for $R_0 \geq 0.9654$ b/u. The rate region resulting from a deterministic time-division schedule, i.e., time-sharing between $(R_0, R_1) = (0.5 \log_2 3, 0)$ b/u and $(R_0, R_1) = (0, \log_2 3))$ b/u, is depicted in Fig. 2.5. It can be seen that the time-sharing region is contained in $\mathcal{R}_1$. We finally point out that the cut-set region $\mathcal{C}_1^\uparrow$ is indeed the capacity region. This is shown in Chapter 3.

2.7. Extension to Other Half-Duplex Networks

Relay cascades are fundamental building blocks in communication networks. The results derived in the previous sections may be instrumental in order to determine the capacity of half-duplex-constrained networks with different topologies.

2.7.1. Broadcast Trees

Consider the tree structured network depicted in Fig. 2.6. The root wishes to multicast information to all other nodes where the intermediate nodes are half-duplex-constrained. We assume noise-free bit pipes (i.e., $q = 1$) and broadcast behavior at nodes with more
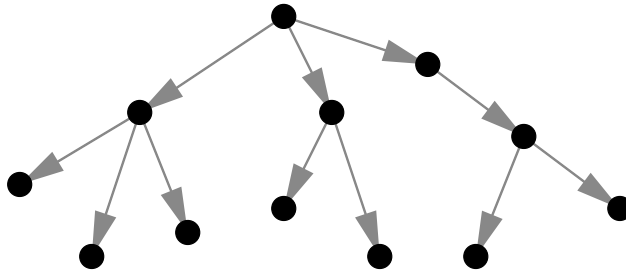
Figure 2.6.: A broadcast tree of depth 3.

than one outgoing edge. The multicast capacity is limited by the capacity of the longest path in the tree, which has four edges. Hence, the multicast capacity of the considered tree is equal to $C_2(1) = 0.7324$ b/u (see Table 2.5), the capacity of a cascade composed of a source, a sink and two half-duplex-constrained relays. More generally, the multicast capacity of any directed tree composed of error-free half-duplex-constrained relays can be obtained using Theorem 2.1 and Theorem 2.2. It approaches

$$C_\infty(q) = \log_2 \left(\frac{1 + \sqrt{4q + 1}}{2} \right) \text{ b/u}$$

as the tree-depth becomes large.

2.7.2. The Butterfly Network

A half-duplex butterfly network [14] is shown in Fig. 2.7. Nodes 1 and 2 intend to multicast information to sink nodes 4 and 5 via both a direct link and a half-duplex-constrained relay node 3. Similar to the previous section, broadcast transmission and bit pipes (i.e., $q = 1$) are assumed. All nodes with two incoming edges behave according to a collision model. To be more precise, a node erases received information if there was a transmission on both incoming links.

By means of network coding (NC) with a bit-wise XOR at the relay, each source node achieves a multicast rate of $\frac{1}{3}$ b/u, i.e., an information flow of $\frac{2}{3}$ b/u is received at each sink node. The well-known strategy is that node 1 broadcasts in the first time slot a binary symbol u_1 to nodes 3 and 4, node 2 broadcasts in the second time slot a binary symbol u_2 to nodes 3 and 5 and, in the third time slot, the relay broadcasts $u_1 \oplus u_2$ to
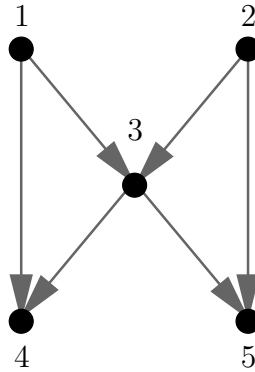
Figure 2.7.: The binary half-duplex butterfly network.

both sink nodes.

The coding strategy introduced in Section 2.4.2 yields a rate of 0.7729 b/u at each sink node. The longest path in the network consists of two edges. Hence, by means of timing, node 1 is able to send information at a rate of $C_1(1) = 0.7729$ b/u to nodes 4 and 5. Similarly, node 2 can send information to nodes 4 and 5 at a rate of $C_1(1) = 0.7729$ b/u. Time-sharing of both source nodes yields a multicast rate of 0.7729 b/u. Decoding at sink nodes 4 and 5 is done as follows. Assume that node 1 is sending information. The sequence received at sink node 4 is a "superposition" of the sequence sent by source node 1 on the direct link $(1, 4)$ and of the relay sequence sent on the edge $(3, 4)$. Due to the timing-strategy, source node 1 and the relay never transmit in the same time slot. Hence, sink node 4 is able to extract the information sent by source node 0 from the received sequence by applying the following protocol. In the very first block, source node 0 broadcasts a message to sink node 4 and the relay while the relay is quiet. Sink node 4 and the relay are able to decode successfully. In the second block, the relay broadcasts the decoded message to nodes 4 and 5 while source node 0 broadcasts a new message to relay node 3 and sink node 4. Since sink node 4 knows the relay strategy and, therefore, the sequence used by the relay for encoding the source message of the previous block, it can determine the new source message sent on the direct link $(1, 4)$ by "subtracting" the relay sequence from the received sequence (which is a superposition of the source- and the relay sequence). Sink node 5 is also able to decode the received relay sequence by applying the rules of the timing strategy introduced in Section 2.4.2. The procedure is repeated in the following blocks and is used in the same way to transmit

information from source node 2 to sink nodes 4 and 5.

2.8. Discussion

The half-duplex constraint is a property common to many wireless networks. In order to overcome the half-duplex constraint, practical transmission protocols split the time of each network node into transmission- and reception-periods, organized in deterministic fashion. This is not optimal from an information theoretic point of view as we demonstrate by means of noise-free relay cascades of various lengths with one or multiple sources. We show that significant rate gains are possible if additional information is encoded by means of timing on the symbol level. On the other hand, timing on the symbol level requires the relays to switch on and off rapidly and to do synchronization among the relays on the symbol level. These requirements can be loosened by sacrificing a bit of the timing rate, i.e., by timing sub-blocks of transmission symbols instead of single transmission symbols. We finally point out that channel model (2.2) and the timing codes introduced in Section 2.4.2 exhibit similarities with the model and the code proposed in [32] for a memory with defective cells. Therein it is assumed that a certain number t of defective cells of a memory composed of n cells always read out a 0, and others always a 1, regardless of what binary symbols are actually stored in them. The objective is to reliably store information under the assumption that the encoder knows the positions and the types of error. Surprisingly, $n - t$ bits can be reliably stored in the memory for large n. The coding scheme is based on the idea that a codebook is generated for each pattern of defective cells. This model resembles our half-duplex relay model in the sense that every time a relay transmits it causes a defective slot in the received sequence while symbols are perfectly received in non-transmission slots. Moreover, by means of cooperation, the sender is aware of the defective slots. The coding approaches we proposed are also based on building up a codebook for every possible pattern of defective slots.

3

Capacity for Half-Duplex Line Networks with Two Sources[†]

3.1. Introduction

In Chapter 2, we considered error-free half-duplex-constrained line networks where a source and a subset of the relays send independent information to the destination. In case of a single source-destination pair, the capacity of cascades and trees of arbitrary depth was established. For the more general case, namely that also the relays are information sources, we stated an achievable rate region in Theorem 2.3. An open question is whether the achievable rate region is the capacity region? So far we know only that the rate region coincides with the cut-set region if the transmission rates of the relay sources fall below certain thresholds. In this chapter, we answer the question for the special case of error-free half-duplex line networks with two sources where either

---

[†]This chapter is based on the following publication [33]: *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, Jun. 13-18 2010, pp. 2393-2397 (together with Gerhard Kramer and Christoph Hausl).

the first or the last relay in the cascade is the second source. In both cases, we establish the capacity region. The achievability scheme presented for the first case builds up on the scheme presented in Chapter 2 and uses the new idea that links with a half-duplex-constrained sink can be interpreted as erasure channels. The achievability scheme for the second case is based on a random coding argument using superposition coding.
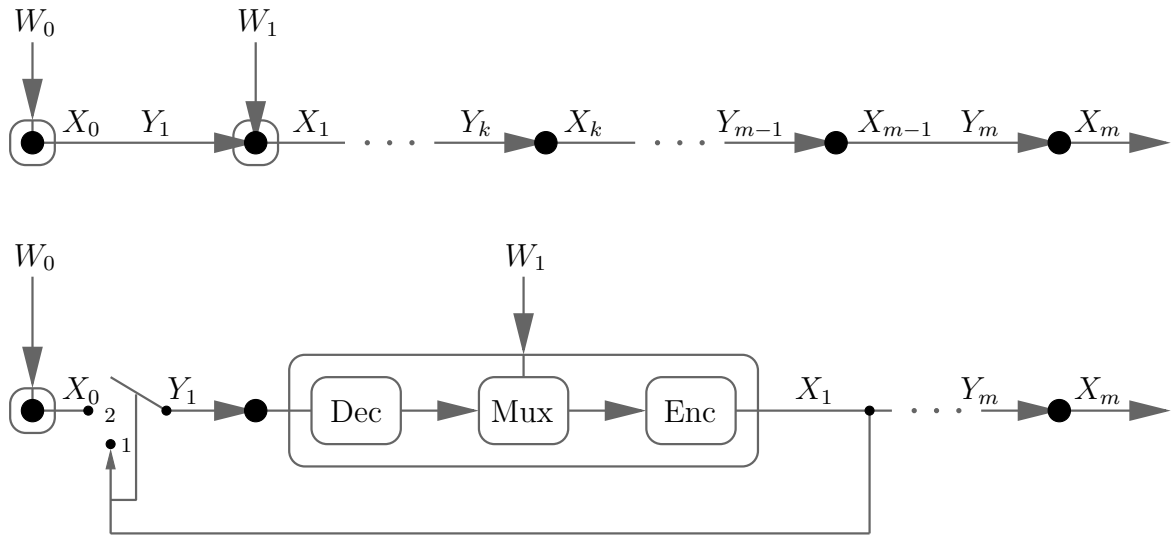
(a) Error-free half-duplex line network where the first two nodes are sources.
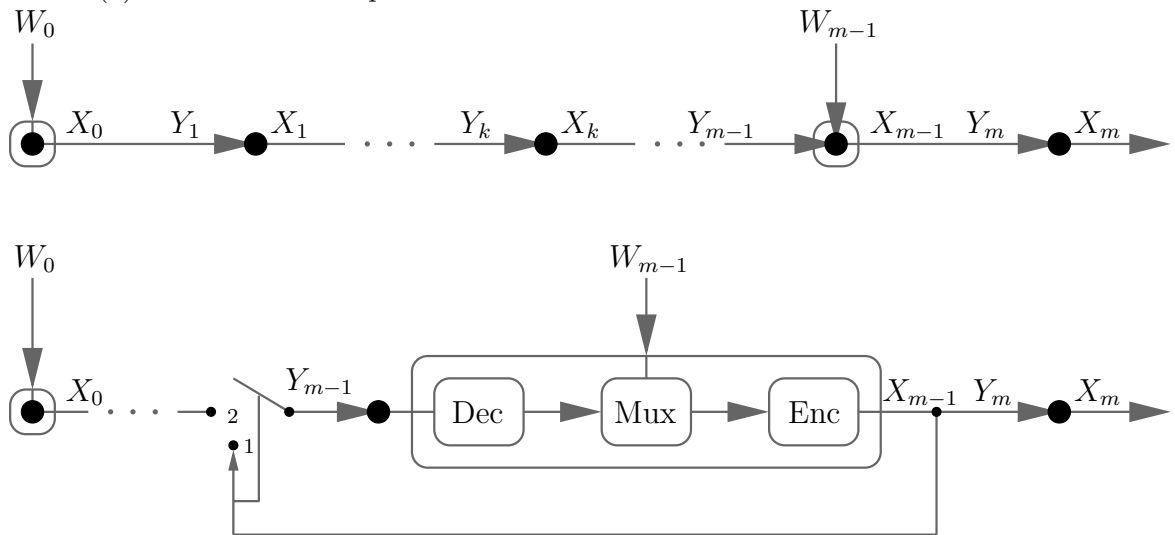
(b) Error-free half-duplex line network where the first node and the second last node are sources.

Figure 3.1.: Error-free half-duplex line networks with two sources.

This chapter is structured as follows. The system model is introduced in Section 3.2. The capacity regions for the first and the second case are derived in Section 3.3 and Section 3.4, respectively. We end with concluding remarks in Section 3.5.

3.2. Network Model

Consider the error-free half-duplex line network with two sources, as depicted in Fig. 3.1, where either the first relay or the last relay is the second source. The system model is a special case of the model introduced in Fig. 3.1. In particular, each half-duplex-constrained relay behaves according to (2.2).

We make the following observation. A network link with a half-duplex-constrained node at the receiving side is effectively an *erasure channel* where the receiver can decide which positions to erase. Moreover, the *erasure probability* equals the probability that the receiving node transmits. The observation can be deduced as follows. If relay i

(a) Relay i transmits.

(b) Relay i listens.
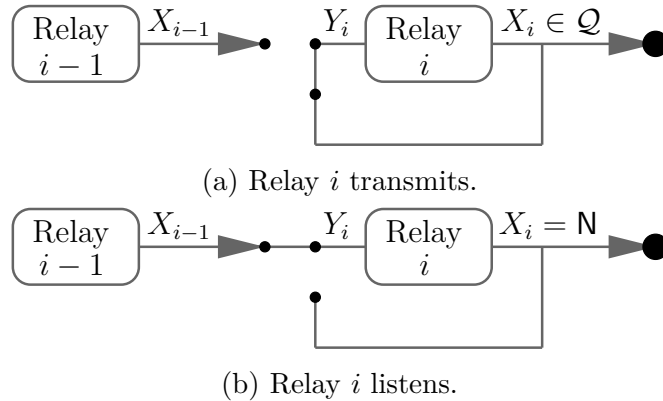
Figure 3.2.: An error-free half-duplex link interpreted as erasure channel.

transmits, it is disconnected from relay $i-1$ due to (2.2), which is depicted in Fig. 3.2a. Therefore, each symbol sent by relay $i-1$ is erased. Since relay i obviously knows its transmission slots, it also knows the erasure locations. If relay i is off, it can perfectly listen to relay $i-1$ via a error-free $(|\mathcal{Q}|+1)$-ary channel, which is depicted in Fig. 3.2b.

The following assumptions regarding the generation and the encoding of information are made. At the beginning of a new block b of n channel uses, source node 0 and relay $i \in \{1, m-1\}$ produce a uniformly and independently drawn message $W_{0;b} \in$

$\left\{1, \ldots, 2^{nR_0}\right\}$ and $W_{i;b} \in \left\{1, \ldots, 2^{nR_i}\right\}$, respectively. Based on the received sequence in block b, sink node m forms the estimates $\hat{w}_{0;b-(m-1)}$ and $\hat{w}_{i;b-(m-1-i)}$ of $W_{0;b-(m-1)}$ and $W_{i;b-(m-1-i)}$. The following encoding functions for the kth time instance are considered:

$$x_{0,k} = f_{0,k}(W_{0;b}) \tag{3.1}$$

$$x_{i,k} = f_{i,k}(W_{i;b}, \mathbf{Y}_i^{k-1}) \tag{3.2}$$

$$x_{l,k} = f_{l,i}(\mathbf{Y}_l^{k-1}) \qquad \text{for all } l \neq \{0, i, m\}. \tag{3.3}$$

3.3. The First Relay is a Source

Theorem 3.1. The capacity region $\mathcal{C}$ of the line network depicted in Fig. 3.1a is

$$\mathcal{C} = \bigcup_{P_{X_0 \cdots X_m}} \left\{ \begin{array}{l} R_0 \leq H(Y_1|X_1) \\ R_0 + R_1 \leq \min_{2 \leq i \leq m} H(Y_i|X_i) \end{array} \right\}. \tag{3.4}$$

It suffices to take the union over all probability distributions $P_{X_0 \cdots X_m}$ of the form

$$P_{X_0} P_{X_1} P_{X_2|X_1} P_{X_3|X_2} \ldots P_{X_m|X_{m-1}} \tag{3.5}$$

with marginal pmf $P_{X_0 X_1}$ as shown in Table 3.1 and marginal pmfs $P_{X_i X_{i+1}}$, $i \in \mathbb{N}$ as shown in Table 2.3a.

Proof. At the end of block $b-1$, node 0 and relay 1 choose new messages $w_{0;b}$ and $w_{1;b}$, which are sent in block b using the sequences $\mathbf{x}_0(w_{0;b})$ and $\mathbf{x}_1(w_{0;b-1}, w_{1;b})$, respectively. Each of the remaining relays forwards a pair of messages from the previous blocks. In particular, relay i sends $\mathbf{x}_i(w_{0;b-i}, w_{1;b-(i-1)})$ in block b, where $2 \leq i \leq m-1$.

Code Construction:

▷ Relay i, $1 \leq i \leq m-1$: The relay nodes apply the same cooperative coding strategy as introduced in Section 2.4.2. In particular, the number of codewords available at relay i to encode a pair of messages $\left(w_{0;b-i}, w_{1;b-(i-1)}\right)$ is upper bounded by

$$|\mathcal{W}_0| \cdot |\mathcal{W}_1| \leq |\mathcal{Q}|^{n_i} \binom{n - n_{i+1}}{n_i}. \tag{3.6}$$

▷ Source node 0: In contrast to the coding strategy introduced in Section 2.4.2, node 0 does not cooperate with node 1. Following the observation made in Section 3.2, it uses an optimal point to point erasure channel code with alphabet $\mathcal{X}$ for encoding $W_{0;b}$. Each symbol received at relay 1 is erased with a probability of $1 - P_{X_1}(\mathsf{N})$.

Achievable Rates: The first link in the cascade is a $|\mathcal{X}|$-ary erasure channel with erasure probability $1 - P_{X_1}(\mathsf{N})$. It is well-known [34, Chapter 8] that the capacity of such a channel equals $P_{X_1}(\mathsf{N}) \log |\mathcal{X}|$ achieved by a uniform input distribution over $\mathcal{X}$. Due to the channel model, we clearly have

$$H(Y_1|X_1) = H(X_0|X_1 = \mathsf{N}) \leq P_{X_1}(\mathsf{N}) \log |\mathcal{X}|, \tag{3.7}$$

with equality if P_{X_0} is the uniform distribution over $|\mathcal{X}|$. Hence, an optimal erasure channel code for the first link satisfies $R_0 = H(Y_1|X_1) - \epsilon$ with $\epsilon \to 0$ as $n \to \infty$. Moreover, independent X_0 and X_1 suffice to achieve equality in (3.7). Further, by the proof of Theorem 2.1, we have

$$|\mathcal{Q}|^{n_i} \binom{n - n_{i+1}}{n_i} \to 2^{nH(Y_{i+1}|X_{i+1})} \qquad \text{as } n \to \infty \tag{3.8}$$

for $1 \leq i \leq m - 1$. Hence, $R_0 + R_1 \leq \min_{2 \leq i \leq m} H(Y_i|X_i)$.

The converse is immediate. By Lemma 2.2, the bounds in (3.4) are the cut-set upper bound (2.35) of the considered network.

Property (3.5) and the pmfs $P_{X_i X_{i+1}}$, $i \in \mathbb{N}_0$, follow from the following consideration. Due to channel model (2.2), $H(Y_i|X_i) = P_{X_i}(\mathsf{N}) H(X_{i-1}|X_i = \mathsf{N})$ for all $1 \leq i \leq m$. Hence, $H(Y_i|X_i)$ is a function of $P_{X_{i-1} X_i}$ and the Markov chain $X_1 - \cdots - X_m$ holds. Further, by the same arguments made in the proof of Theorem 2.1, we may assume marginal pmfs $P_{X_i X_{i+1}}$ as shown in Table 2.3a for all $i \in \mathbb{N}$. For $P_{X_0 X_1}$, we may assume the distribution shown in Table 3.1 because it achieves the upper bound in (3.7). The probability masses were chosen such that X_0 and X_1 are independent and P_{X_0} is uniform, thus matching the properties of X_0 and X_1. ∎

Remark 3.1. An application of Theorem 3.1 to the setup considered in Example 2.5

X_0 \ X_1	0	$\cdots$	$q-1$	N
0	$\bar{p}_1/[q(q+1)]$	$\cdots$	$\bar{p}_1/[q(q+1)]$	$p_1/(q+1)$
$\vdots$				
$q-1$				
N	$\bar{p}_1/[q(q+1)]$	$\cdots$	$\bar{p}_1/[q(q+1)]$	$p_1/(q+1)$

Table 3.1.: Optimum Pmf $P_{X_0 X_1}$ for Source Node 0 using an Erasure Code.

shows that the achievable rate region (2.48) is not the capacity region if $R_0 \leq R_0^\star$. However, by letting source node 0 use an independent erasure code, as we did in Theorem 3.1, we get rid of constraint (2.44). The reason is that source node 0 and relay 1 do not cooperate anymore, which enables the relay to encode own information also by means of transmission patterns. Hence, the capacity region is equal to the cut-set region $\mathcal{C}_1^\uparrow$:

$$
R_1 = \begin{cases}
\log(q+1) - R_0, \ 0 \leq R_0 < \dfrac{1}{q+1}\log(q+1) \\[3mm]
\left(1 - \dfrac{R_0}{\log(q+1)}\right)\log q + H\left(\dfrac{R_0}{\log(q+1)}\right) - R_0, \ \dfrac{1}{q+1}\log(q+1) \leq R_0 \leq C_1(q).
\end{cases}
$$

The capacity region for $q = 2$ is depicted in Fig. 2.5.

3.4. The Last Relay is a Source

In this section, we derive the capacity region of the line network shown in Fig. 3.1b. Since the achievability proof is based on a random coding argument, we first introduce a couple of tools that are widely used in information theory.

Definition 3.1. [35, Chapter 2] The *ϵ-typical set* $\mathcal{T}_\epsilon^{(n)}(X)$ is the set of n-sequences $\mathbf{x}$, which satisfy

$$
\mathcal{T}_\epsilon^{(n)}(X) = \{\mathbf{x} : |\pi(x|\mathbf{x}) - P_X(x)| \leq \epsilon P_X(x) \text{ for all } x \in \mathcal{X}\}, \tag{3.10}
$$

where $\pi(x|\mathbf{x})$ is the type of symbol x based on sequence $\mathbf{x}$, i.e.,

$$
\pi(x|\mathbf{x}) = \frac{|\{i : x_i = x\}|}{n} \quad \text{for } x \in \mathcal{X}.
$$

The notion of a ϵ-typical set can be extended in a natural way to random vectors, for example (U, X, Y), by considering the random vector as a single random variable. The corresponding notation is $\mathcal{T}_\epsilon^{(n)}(U, X, Y)$. The next lemma is used in many achievability proofs of multiuser theorems.

Lemma 3.1 (Packing Lemma [35])**.** Let $(U, X, Y) \sim P_{UXY}$. Let $(\tilde{\mathbf{U}}, \tilde{\mathbf{Y}})$ be a pair of arbitrarily distributed random n-sequences, not necessarily distributed according to $\prod_{i=1}^n P_{UY}(\tilde{u}_i, \tilde{y}_i)$. Let $\mathbf{X}(w)$, $w \in \mathcal{W}$, where $|\mathcal{W}| \leq 2^{nR}$, be random n-sequences, each distributed according to $\prod_{i=1}^n P_{X|U}(x_i|\tilde{u}_i)$. Further assume that $\mathbf{X}(w)$, $w \in \mathcal{W}$, is pairwise conditionally independent of $\tilde{\mathbf{Y}}$ given $\tilde{\mathbf{U}}$, but is arbitrarily dependent on other $\mathbf{X}(w)$ sequences. Then, there exists $\delta(\epsilon)$ that tends to zero as $\epsilon \to 0$ such that

$$\lim_{n \to \infty} \mathbb{P}\left\{ \left(\tilde{\mathbf{U}}, \mathbf{X}(w), \tilde{\mathbf{Y}}\right) \in \mathcal{T}_\epsilon^{(n)}(U, X, Y) \text{ for some } w \in \mathcal{W} \right\} = 0$$

if $R < I(X; Y|U) - \delta(\epsilon)$.

In the following definition, the standard notion of a *random code* is tailored to the problem setup considered here.

Definition 3.2. A $\left(2^{nR_0}, 2^{nR_{m-1}}, n\right)$ *code* for the line network shown in Fig. 3.1b consists of

$\triangleright$ two message sets $\mathcal{W}_0 = \{1, \ldots, 2^{nR_0}\}$ and $\mathcal{W}_{m-1} = \{1, \ldots, 2^{nR_{m-1}}\}$ where

  – we split W_0 into B sub-blocks $W_{0;b}$, $b = 1, 2, \ldots, B$, that each take on 2^{nR_0} values;

  – we split W_{m-1} into B sub-blocks $W_{m-1;b}$, $b = 1, 2, \ldots, B$, that each take on $2^{nR_{m-1}}$ values.

$\triangleright$ m encoders where

  – the encoder at node i, $0 \leq i \leq m - 2$, assigns a codeword $\mathbf{x}_i\left(w_{0;b-[i;m-1]}\right)$ of length n to the messages $\left(w_{0;b-[i;m-1]}\right) \stackrel{def}{=} \left(w_{0;b-i}, \ldots, w_{0;b-(m-1)}\right) \in \mathcal{W}_0^{m-i}$ for transmission in block b;

  – the encoder at node $m - 1$ assigns a codeword $\mathbf{x}_{m-1}\left(w_{0;b-(m-1)}, w_{m-1;b}\right)$ of length n to the messages $\left(w_{0;b-(m-1)}, w_{m-1;b}\right) \in \mathcal{W}_0 \times \mathcal{W}_{m-1}$ for transmission in block b.

▷ m decoders where

– the decoder at node i, $1 \le i \le m-1$, assigns an estimate $\hat{w}_{0;b-(i-1)} \in \mathcal{W}_0$ or an error message to the received n-sequence $\mathbf{y}_i(b)$;

– the decoder at node m assigns an estimate $\left(\hat{w}_{0;b-(m-1)}, \hat{w}_{m-1;b}\right) \in \mathcal{W}_0 \times \mathcal{W}_{m-1}$ or an error message to the received n-sequence $\mathbf{y}_m(b)$.

Definition 3.3. The *average probability of error* in block b

▷ at node i, $1 \le i \le m-1$, is

$$P_{i;b}^{(n)} = \mathbb{P}\left\{\hat{W}_{0;b-(i-1)} \ne W_{0;b-(i-1)}\right\};$$

▷ at node m is

$$P_{m;b}^{(n)} = \mathbb{P}\left\{\left(\hat{W}_{0;b-(m-1)}, \hat{W}_{m-1;b}\right) \ne \left(W_{0;b-(m-1)}, W_{m-1;b}\right)\right\}.$$

A rate pair (R_0, R_{m-1}) is *achievable* if there exists a sequence of $\left(2^{nR_0}, 2^{nR_{m-1}}, n\right)$ codes such that

$$\lim_{n\to\infty} P_{i;b}^{(n)} = 0$$

for all $i = 1, \dots, m$ and $b = 1, \dots, B$.

Theorem 3.2. The capacity region $\mathcal{C}$ of the line network depicted Fig. 3.1b is

$$\mathcal{C} = \bigcup_{P_{X_0 \dots X_{m-1}}} \left\{ \begin{array}{l} R_0 \le \min\left\{\min_{1 \le i \le m-2} H\left(Y_i|X_i\right), H\left(Y_{m-1}|UX_{m-1}\right)\right\} \\ R_{m-1} \le H(Y_m|U) \\ R_0 + R_{m-1} \le H(Y_m) \end{array} \right\}.$$

The union is over all probability distributions $P_{X_0 \dots X_{m-1}}$ of the form

$$P_{X_0} P_{X_1} P_{X_2|X_1} \dots P_{X_{m-2}|X_{m-3}} P_{U|X_{m-2}} P_{X_{m-1}|U}, \tag{3.11}$$

where $|U| \le |\mathcal{X}_{m-2}| \cdot |\mathcal{X}_{m-1}| + 2$.

3.4.1. Achievability Proof for Theorem 3.2

Random codebook generation:

▷ Fix a pmf $P_{X_0} P_{X_1} P_{X_2|X_1} \ldots P_{X_{m-2}|X_{m-3}} P_{U|X_{m-2}} P_{X_{m-1}|U}$.

▷ Randomly and independently generate 2^{nR_0} sequences $\mathbf{u}\left(w_{0;b-(m-1)}\right)$ of length n, $w_{0;b-(m-1)} \in \{1, \ldots, 2^{nR_0}\}$, each according to $\prod_{l=1}^n P_U(u_l)$.

▷ *Codebook at node $m-1$:* For each $\mathbf{u}\left(w_{0;b-(m-1)}\right)$, randomly and independently generate $2^{nR_{m-1}}$ sequences $\mathbf{x}_{m-1}\left(w_{0;b-(m-1)}, w_{m-1;b}\right)$, $w_{m-1;b} \in \{1, \ldots, 2^{nR_{m-1}}\}$, each according to $\prod_{l=1}^n P_{X_{m-1}|U}(x_{m-1,l}|u_l)$ (see Fig. 3.3a).

▷ *Codebook at node $m-2$:* For each $\mathbf{u}\left(w_{0;b-(m-1)}\right)$, randomly and independently generate 2^{nR_0} sequences $\mathbf{x}_{m-2}\left(w_{0;b-(m-2)}, w_{0;b-(m-1)}\right)$, $w_{0;b-(m-2)} \in \{1, \ldots, 2^{nR_0}\}$, each according to $\prod_{l=1}^n P_{X_{m-2}|U}(x_{m-2,l}|u_l)$ (see Fig. 3.3b).

▷ *Codebook at node i, $0 \le i < m-2$:* For each $\mathbf{x}_{i+1}\left(w_{0;b-[i+1;m-1]}\right)$, randomly and independently generate 2^{nR_0} sequences $\mathbf{x}_i\left(w_{0;b-[i;m-1]}\right)$, $w_{0;b-i} \in \{1, \ldots, 2^{nR_0}\}$, each according to $\prod_{l=1}^n P_{X_i|X_{i+1}}(x_{i,l}|x_{i+1,l})$ (see Fig. 3.3c).

Encoding: To send $w_{0;b-i}$ in block b, node i transmits $\mathbf{x}_i\left(w_{0;b-[i;m-1]}\right)$, $0 \le i \le m-2$ based on the previously sent messages $\{w_{0;b-[i+1;m-1]}\}$. Similarly, to send $w_{0;b-(m-1)}$ and $w_{m-1;b}$ in block b, node $m-1$ transmits $\mathbf{x}_{m-1}\left(w_{0;b-(m-1)}, w_{m-1;b}\right)$.

Decoding: The decoder at node i, $1 \le i \le m-2$, declares at the end of block b that $\hat{w}_{0;b-(i-1)}$ is sent if it is the unique message such that

$$\left(\mathbf{x}_{i-1}\left(\hat{w}_{0;b-(i-1)}, w_{0;b-[i;m-1]}\right), \mathbf{x}_i\left(w_{0;b-[i;m-1]}\right), \mathbf{y}_i(b)\right) \in \mathcal{T}_\epsilon^{(n)}(X_{i-1}, X_i, Y_i).$$

The decoder at node $m-1$ declares at the end of block b that $\hat{w}_{0;b-(m-2)}$ is sent if it is the unique message such that

$$\left(\mathbf{u}\left(w_{0;b-(m-1)}\right), \mathbf{x}_{m-2}\left(\hat{w}_{0;b-(m-2)}, w_{0;b-(m-1)}\right), \mathbf{x}_{m-1}\left(w_{0;b-(m-1)}, \hat{w}_{m-1;b}\right), \mathbf{y}_{m-1}(b)\right)$$
$$\in \mathcal{T}_\epsilon^{(n)}(U, X_{m-2}, X_{m-1}, Y_{m-1}).$$

(a) Random codebook construction for node $m-1$.

(b) Random codebook construction for node $m-2$.

(c) Random codebook construction for node i, $0 \leq i < m-2$.
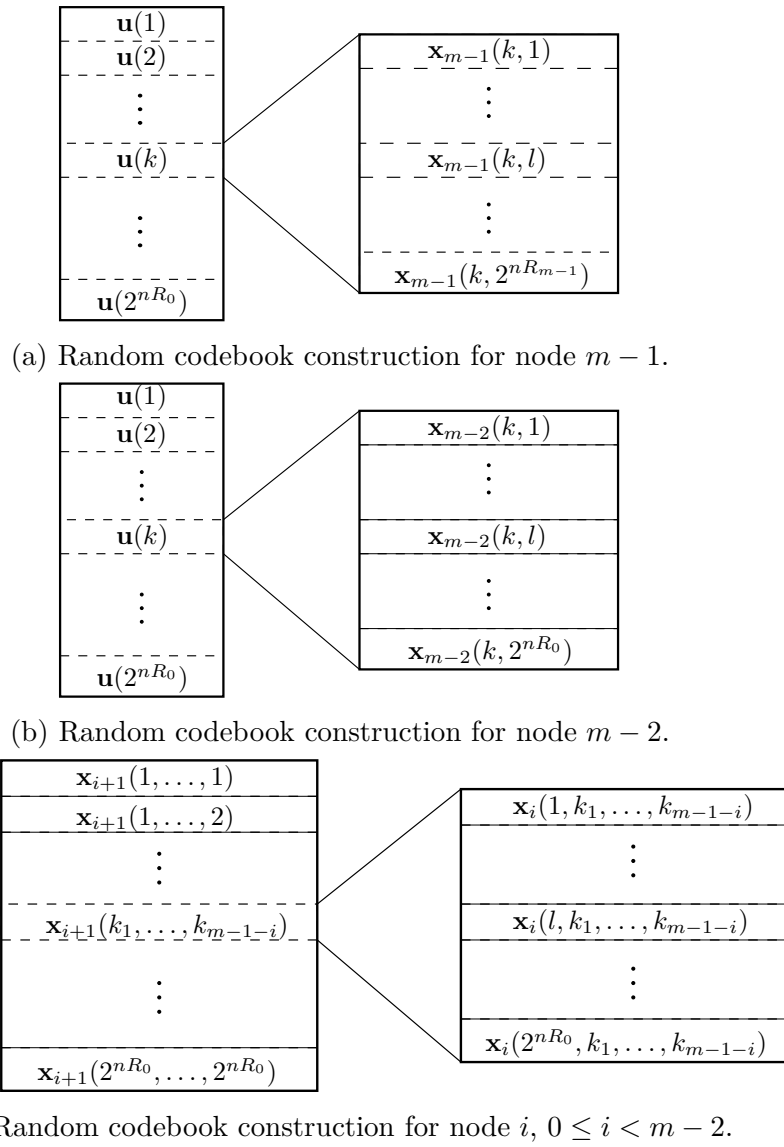
Figure 3.3.: Random codebook construction for the network depicted in Fig. 3.1b.

The decoder at sink node m declares at the end of block b that $\left(\hat{w}_{0;b-(m-1)}, \hat{w}_{m-1;b}\right)$ is sent if they are the unique messages such that

$$\left(\mathbf{u}\left(\hat{w}_{0;b-(m-1)}\right), \mathbf{x}_{m-1}\left(\hat{w}_{0;b-(m-1)}, \hat{w}_{m-1;b}\right), \mathbf{y}_m(b)\right) \in \mathcal{T}_\epsilon^{(n)}(U, X_{m-1}, Y_m).$$

Analysis of the probabilities of error: We assume without loss of generality that $\left(W_{0;b-(m-1)}, \ldots, W_{0;b}, W_{m-1;b}\right) = (1, \ldots, 1, 1)$. Moreover, we assume that all decoding steps in previous blocks were successful. Then the decoder at node i, $1 \le i \le m-2$, makes an error if and only if one or both of the following events occur:

$$\mathcal{E}_{i,1}^{(n)} = \left\{ (\mathbf{X}_{i-1}(1, \ldots, 1), \mathbf{X}_i(1, \ldots, 1), \mathbf{Y}_i(b)) \notin \mathcal{T}_\epsilon^{(n)}(X_{i-1}, X_i, Y_i) \right\},$$

$$\mathcal{E}_{i,2}^{(n)} = \left\{ \left(\mathbf{X}_{i-1}\left(w_{0;b-(i-1)}, 1, \ldots, 1\right), \mathbf{X}_i(1, \ldots, 1), \mathbf{Y}_i(b)\right) \in \mathcal{T}_\epsilon^{(n)}(X_{i-1}, X_i, Y_i) \right\}$$

for some $w_{0;b-(i-1)} \ne 1$. The average probability of error for the decoder in block b at node i is upper bounded as

$$P_{i;b}^{(n)} \le \mathbb{P}\left(\mathcal{E}_{i,1}^{(n)}\right) + \mathbb{P}\left(\mathcal{E}_{i,2}^{(n)}\right).$$

By the law of large numbers, the first term $\mathbb{P}\left(\mathcal{E}_{i,1}^{(n)}\right)$ goes to zero as $n \to \infty$. Regarding the second term, observe that $\mathbf{X}_{i-1}\left(w_{0;b-(i-1)}, 1, \ldots, 1\right)$ is conditionally independent of $(\mathbf{X}_{i-1}(1, 1, \ldots, 1), \mathbf{Y}_i(b))$ given $\mathbf{X}_i(1, \ldots, 1)$ for $w_{0;b-(i-1)} \ne 1$ and is distributed according to $\prod_{k=1}^n P_{X_{i-1}|X_i}(x_{i-1,k}|x_{i,k}(1, \ldots, 1))$. Hence, by Lemma 3.1, $\mathbb{P}\left(\mathcal{E}_{i,2}^{(n)}\right)$ tends to zero as $n \to \infty$ if $R_0 < I(X_{i-1}; Y_i|X_i) - \delta(\epsilon) = H(Y_i|X_i) - \delta(\epsilon)$, where the equality follows from channel model (2.2).

Next, the average probability of error for the decoder at node $m-1$ is analyzed. The decoder makes an error if and only if one or both of the following events occur:

$$\mathcal{E}_{m-1,1}^{(n)} = \left\{ (\mathbf{U}(1), \mathbf{X}_{m-2}(1,1), \mathbf{X}_{m-1}(1,1), \mathbf{Y}_{m-1}(b)) \notin \mathcal{T}_\epsilon^{(n)}(U, X_{m-2}, X_{m-1}, Y_{m-1}) \right\},$$

$$\mathcal{E}_{m-1,2}^{(n)} = \left\{ \left(\mathbf{U}(1), \mathbf{X}_{m-2}\left(w_{0;b-(m-2)}, 1\right), \mathbf{X}_{m-1}(1,1), \mathbf{Y}_{m-1}(b)\right)\right.$$
$$\left. \in \mathcal{T}_\epsilon^{(n)}(U, X_{m-2}, X_{m-1}, Y_{m-1}) \right\}$$

for some $w_{0;b-(m-2)} \neq 1$. The average probability of error for the decoder in block b at node $m-1$ is upper bounded as

$$P_{m-1;b}^{(n)} \leq \mathbb{P}\left(\mathcal{E}_{m-1,1}^{(n)}\right) + \mathbb{P}\left(\mathcal{E}_{m-1,2}^{(n)}\right).$$

By the law of large numbers, the first term $\mathbb{P}\left(\mathcal{E}_{m-1,1}^{(n)}\right)$ goes to zero as $n \to \infty$. Regarding the second term, observe that $\mathbf{X}_{m-2}\left(w_{0;b-(m-2)}, 1\right)$ is conditionally independent of $\left(\mathbf{X}_{m-2}\left(1, 1\right), \mathbf{Y}_{m-1}(b)\right)$ given $\mathbf{U}\left(1\right)$ and $\mathbf{X}_{m-1}\left(1, 1\right)$ for $w_{0;b-(m-2)} \neq 1$ and is distributed according to $\prod_{k=1}^{n} P_{X_{m-2}|U}\left(x_{m-2,k}|u_k(1)\right)$. Hence, by Lemma 3.1, $\mathbb{P}\left(\mathcal{E}_{m-1,2}^{(n)}\right)$ tends to zero as $n \to \infty$ if $R_0 < I(X_{m-2}; Y_{m-1}|UX_{m-1}) - \delta(\epsilon) = H(Y_{m-1}|UX_{m-1}) - \delta(\epsilon)$, where the equality follows from channel model (2.2).

Finally, the average probability of error for the decoder at sink node m is analyzed. The decoding procedure yields an error if and only if one or both of the following events occur:

$$\mathcal{E}_{m,1}^{(n)} = \left\{ \left(\mathbf{U}\left(1\right), \mathbf{X}_{m-1}\left(1, 1\right), \mathbf{Y}_m(b)\right) \notin \mathcal{T}_{\epsilon}^{(n)}(U, X_{m-1}, Y_m) \right\}$$

and

$$\mathcal{E}_{m,2}^{(n)} = \left\{ \left(\mathbf{U}\left(w_{0;b-(m-1)}\right), \mathbf{X}_{m-1}\left(w_{0;b-(m-1)}, 1\right), \mathbf{Y}_m(b)\right) \in \mathcal{T}_{\epsilon}^{(n)}(U, X_{m-1}, Y_m) \right\}$$

for some $w_{0;b-(m-1)} \neq 1$. Further,

$$\mathcal{E}_{m,3}^{(n)} = \left\{ \left(\mathbf{U}\left(1\right), \mathbf{X}_{m-1}\left(1, w_{m-1;b}\right), \mathbf{Y}_m(b)\right) \in \mathcal{T}_{\epsilon}^{(n)}(U, X_{m-1}, Y_m) \right\}$$

for some $w_{m-1;b} \neq 1$, and

$$\mathcal{E}_{m,4}^{(n)} = \left\{ \left(\mathbf{U}\left(w_{0;b-(m-1)}\right), \mathbf{X}_{m-1}\left(w_{0;b-(m-1)}, w_{m-1;b}\right), \mathbf{Y}_m(b)\right) \in \mathcal{T}_{\epsilon}^{(n)}(U, X_{m-1}, Y_m) \right\}$$

for some $w_{0;b-(m-1)} \neq 1$, $w_{m-1;b} \neq 1$. The average probability of error for the decoder in block b at sink node m is upper bounded as

$$P_{m;b}^{(n)} \leq \mathbb{P}\left(\mathcal{E}_{m,1}^{(n)}\right) + \mathbb{P}\left(\mathcal{E}_{m,2}^{(n)}\right) + \mathbb{P}\left(\mathcal{E}_{m,3}^{(n)}\right) + \mathbb{P}\left(\mathcal{E}_{m,4}^{(n)}\right).$$

By the law of large numbers, the first term $\mathbb{P}\left(\mathcal{E}_{m,1}^{(n)}\right)$ goes to zero as $n \to \infty$. For the second term, note that $\left(\mathbf{U}\left(w_{0;b-(m-1)}\right), \mathbf{X}_{m-1}\left(w_{0;b-(m-1)}, 1\right)\right)$ is independent of $(\mathbf{U}(1), \mathbf{X}_{m-1}(1,1), \mathbf{Y}_m(b))$ for $w_{0;b-(m-1)} \neq 1$. Thus, by Lemma 3.1, $\mathbb{P}\left(\mathcal{E}_{m,2}^{(n)}\right)$ tends to zero as $n \to \infty$ if $R_0 < I(UX_{m-1}; Y_m) - \delta(\epsilon) = H(Y_m) - \delta(\epsilon)$, where the equality follows from channel model (2.2). For the third term, observe that $\mathbf{X}_{m-1}(1, w_{m-1;b})$ is conditionally independent of $(\mathbf{X}_{m-1}(1,1), \mathbf{Y}_m(b))$ given $\mathbf{U}(1)$ for $w_{m-1;b} \neq 1$ and is distributed according to $\prod_{k=1}^n P_{X_{m-1}|U}(x_{m-1,k}|u_k(1))$. Hence, by Lemma 3.1, $\mathbb{P}\left(\mathcal{E}_{m,3}^{(n)}\right)$ tends to zero as $n \to \infty$ if $R_{m-1} < I(X_{m-1}; Y_m|U) - \delta(\epsilon) = H(Y_m|U) - \delta(\epsilon)$, where the equality follows from channel model (2.2). Finally, regarding the fourth term, observe that the tuple $\left(\mathbf{U}\left(w_{0;b-(m-1)}\right), \mathbf{X}_{m-1}\left(w_{0;b-(m-1)}, w_{m-1;b}\right)\right)$ is independent of $(\mathbf{U}(1), \mathbf{X}_{m-1}(1,1), \mathbf{Y}_m(b))$ for $w_{0;b-(m-1)} \neq 1$, $w_{m-1;b} \neq 1$. Hence, by Lemma 3.1, $\mathbb{P}\left(\mathcal{E}_{m,4}^{(n)}\right)$ tends to zero as $n \to \infty$ if $R_0 + R_1 < I(UX_{m-1}; Y_m) - \delta(\epsilon) = H(Y_m) - \delta(\epsilon)$, where the equality follows from channel model (2.2).

3.4.2. Converse for Theorem 3.2

Consider a $\left(2^{nR_0}, 2^{nR_{m-1}}, n\right)$ code with $\lim_{n\to\infty} P_{i;b}^{(n)} = 0$ for all $1 \leq i \leq m$ and $b = 1, 2, \ldots, B$. By Fano's inequality

$$H\left(W_{0;b-(i-1)}|\mathbf{Y}_i^n\right) \leq nR_0 P_{i;b}^{(n)} + 1 = n\epsilon_{i;b}^{(n)}, \quad \text{for all } 1 \leq i \leq m-1 \tag{3.12}$$

and

$$H\left(W_{0;b-(m-1)}W_{m-1;b}|\mathbf{Y}_m^n\right) \leq n(R_0 + R_{m-1})P_{m;b}^{(n)} + 1 = n\epsilon_{m;b}^{(n)} \tag{3.13}$$

where $\epsilon_{i;b}^{(n)}$ tends to zero as $n \to \infty$ for all $1 \leq i \leq m$. For $1 \leq i \leq m-2$, we have

$$nR_0 = H\left(W_{0;b-(i-1)}\right) \tag{3.14}$$

$$= H\left(W_{0;b-(i-1)}|W_{m-1;b}\right) \tag{3.15}$$

$$= I\left(W_{0;b-(i-1)}; \mathbf{Y}_i^n|W_{m-1;b}\right) + H\left(W_{0;b-(i-1)}|\mathbf{Y}_i^n W_{m-1;b}\right)$$

$$\leq I\left(W_{0;b-(i-1)}; \mathbf{Y}_i^n|W_{m-1;b}\right) + n\epsilon_{i;b}^{(n)} \tag{3.16}$$

$$= \sum_{l=1}^n I\left(W_{0;b-(i-1)}; Y_{i,l}|W_{m-1;b}\mathbf{Y}_i^{l-1}\right) + n\epsilon_{i;b}^{(n)} \tag{3.17}$$

$$= \sum_{l=1}^{n} I\left(W_{0;b-(i-1)}; Y_{i,l} | W_{m-1;b} \mathbf{Y}_i^{l-1} \mathbf{X}_i^l\right) + n\epsilon_{i;b}^{(n)} \tag{3.18}$$

$$\leq \sum_{l=1}^{n} I\left(W_{0;b-(i-1)} X_{i-1,l}; Y_{i,l} | W_{m-1;b} \mathbf{Y}_i^{l-1} \mathbf{X}_i^l\right) + n\epsilon_{i;b}^{(n)} \tag{3.19}$$

$$\leq \sum_{l=1}^{n} I\left(X_{i-1,l}; Y_{i,l} | X_{i,l}\right) + n\epsilon_{i;b}^{(n)} \tag{3.20}$$

$$= nI\left(X_{i-1}; Y_i | X_i Q\right) + n\epsilon_{i;b}^{(n)} \tag{3.21}$$

$$\leq nH\left(Y_i | X_i\right) + n\epsilon_{i;b}^{(n)}, \tag{3.22}$$

where[2]

(3.14) follows because $W_{0;b-(i-1)}$ is uniformly distributed;

(3.15) follows from the independence of $W_{0;b-(i-1)}$ and $W_{m-1;b}$;

(3.16) follows from (3.12) and the property that conditioning does not increase entropy;

(3.17) follows from the chain rule for mutual information;

(3.18) follows from (3.3), i.e., $\mathbf{X}_i^l$ is a function of $\mathbf{Y}_i^{l-1}$ for all $1 \leq i \leq m - 2$;

(3.19) follows from the chain rule and the non-negativity of mutual information;

(3.20) follows because $Y_{i,l}$ is a function of $X_{i-1,l}$ and $X_{i,l}$;

(3.21) follows by defining Q to be a time-sharing random variable with $Y_i \stackrel{def}{=} Y_{i,Q}$, $X_{i-1} \stackrel{def}{=} X_{i-1,Q}$ and $X_i \stackrel{def}{=} X_{i,Q}$;

(3.22) follows by expanding mutual information and using the fact that conditioning does not increase entropy.

In order to bound R_0 for $i = m - 1$, we start with (3.19). Before doing so, we remark that (3.18) holds since $\mathbf{X}_{m-1}^l$ is a function of $\mathbf{Y}_{m-1}^{l-1}$ and $W_{m-1;b}$ due to (3.2). We have

$$nR_0 \leq \sum_{l=1}^{n} I\left(W_{0;b-(m-2)} X_{m-2,l}; Y_{m-1,l} | W_{m-1;b} \mathbf{Y}_{m-1}^{l-1} \mathbf{X}_{m-1}^l\right) + n\epsilon_{m-1;b}^{(n)}$$

---

[2]One could derive (3.22) directly from (3.18) due to the deterministic network model. Because of Section 3.4.4, we introduce a couple of additional steps.

$$\leq \sum_{l=1}^{n} H\left(Y_{m-1,l}|V_l X_{m-1,l}\right) + n\epsilon_{m-1;b}^{(n)} \tag{3.23}$$

$$= nH\left(Y_{m-1}|U X_{m-1}\right) + n\epsilon_{m-1;b}^{(n)}, \tag{3.24}$$

where

(3.23) follows from the deterministic network model and because conditioning does not increase entropy; further $V_l \stackrel{def}{=} \left(\mathbf{X}_{m-1}^{l-1}, \mathbf{Y}_{m-1}^{l-1}\right)$;

(3.24) follows by defining Q to be a time-sharing random variable with $U \stackrel{def}{=} (V_Q, Q)$, $Y_{m-1} \stackrel{def}{=} Y_{m-1,Q}$ and $X_{m-1} \stackrel{def}{=} X_{m-1,Q}$.

Next, note from (3.13) that $H\left(W_{m-1;b}|\mathbf{Y}_m^n\right) \leq n\epsilon_{m;b}^{(n)}$. Hence, we have

$$nR_{m-1} = H\left(W_{m-1;b}\right)$$

$$= I\left(W_{m-1;b}; \mathbf{Y}_m^n\right) + H\left(W_{m-1;b}|\mathbf{Y}_m^n\right)$$

$$\leq I\left(W_{m-1;b}; \mathbf{Y}_m^n\right) + n\epsilon_{m;b}^{(n)}$$

$$= \sum_{l=1}^{n} I\left(W_{m-1;b}; Y_{m,l}|W_0 \mathbf{Y}_m^{l-1}\right) + n\epsilon_{m;b}^{(n)} \tag{3.25}$$

$$= \sum_{l=1}^{n} H\left(Y_{m,l}|W_0 \mathbf{X}_{m-1}^{l-1} \mathbf{Y}_m^{l-1}\right) + n\epsilon_{m;b}^{(n)} \tag{3.26}$$

$$= \sum_{l=1}^{n} H\left(Y_{m,l}|W_0 \mathbf{X}_{m-1}^{l-1} \mathbf{Y}_{m-1}^{l-1} \mathbf{Y}_m^{l-1}\right) + n\epsilon_{m;b}^{(n)} \tag{3.27}$$

$$\leq \sum_{l=1}^{n} H\left(Y_{m,l}|V_l\right) + n\epsilon_{m;b}^{(n)} \tag{3.28}$$

$$= nH\left(Y_m|U\right) + n\epsilon_{m;b}^{(n)}, \tag{3.29}$$

where

(3.25) follows from the chain rule for mutual information and the independence of W_0 and $W_{m-1;b}$;

(3.26) follows because W_0 (see Definition 3.2) and $W_{m-1;b}$ determine $Y_{m,l}$; further, $\mathbf{Y}_m^{l-1} = \mathbf{X}_{m-1}^{l-1}$;

(3.27) follows because W_0 and $\mathbf{X}_{m-1}^{l-1}$ fully determine $\mathbf{Y}_{m-1}^{l-1}$;

(3.28) follows by $V_l \overset{def}{=} \left(\mathbf{X}_{m-1}^{l-1}, \mathbf{Y}_{m-1}^{l-1} \right)$ and the fact that conditioning does not increase entropy;

(3.29) follows by defining Q to be a time-sharing random variable with $U \overset{def}{=} (V_Q, Q)$ and $Y_m \overset{def}{=} Y_{m,Q}$.

Concerning the sum-rate, we obtain[3]

$$nR_0 + nR_{m-1} = H\left(W_{0;b-(m-1)}W_{m-1;b} \right) \tag{3.30}$$

$$= I\left(W_{0;b-(m-1)}W_{m-1;b}; \mathbf{Y}_m^n \right) + H\left(W_{0;b-(m-1)}W_{m-1;b}|\mathbf{Y}_m^n \right)$$

$$\leq I\left(W_{0;b-(m-1)}W_{m-1;b}; \mathbf{Y}_m^n \right) + n\epsilon_{m;b}^{(n)} \tag{3.31}$$

$$= \sum_{l=1}^{n} I\left(W_{0;b-(m-1)}W_{m-1;b}; Y_{m,l}|\mathbf{Y}_m^{l-1} \right) + n\epsilon_{m;b}^{(n)} \tag{3.32}$$

$$\leq \sum_{l=1}^{n} I\left(W_{0;b-(m-1)}W_{m-1;b}X_{m-1,l}; Y_{m,l}|\mathbf{Y}_m^{l-1} \right) + n\epsilon_{m;b}^{(n)} \tag{3.33}$$

$$\leq \sum_{l=1}^{n} I\left(X_{m-1,l}; Y_{m,l} \right) + n\epsilon_{m;b}^{(n)} \tag{3.34}$$

$$= nI\left(X_{m-1}; Y_m|Q \right) + n\epsilon_{m;b}^{(n)} \tag{3.35}$$

$$\leq nH\left(Y_m \right) + n\epsilon_{m;b}^{(n)}, \tag{3.36}$$

where

(3.30) follows because $W_{0;b-(m-1)}$ and $W_{m-1;b}$ are independent and uniformly distributed;

(3.31) follows using (3.13);

(3.32) and (3.33) follow from the chain rule for mutual information;

(3.34) follows from the Markov chain $\left(W_{0;b-(m-1)}, W_{m-1;b} \right) - X_{m-1,l} - Y_{m,l}$ and the fact that conditioning does not increase entropy;

(3.35) follows by defining Q to be a time-sharing random variable and $Y_m \overset{def}{=} Y_{m,Q}$, $X_{m-1} \overset{def}{=} X_{m-1,Q}$;

---

[3]One could derive (3.36) directly from (3.32) due to the deterministic network model. Because of Section 3.4.4, we introduce a couple of additional steps.

(3.36) follows by expanding mutual information and using the fact that conditioning does not increase entropy.

To complete the proof of the converse, we need to check (3.11). Recall (3.2), namely that $X_{m-1,l}$ is a function of $\mathbf{Y}_{m-1}^{l-1}$ and W_{m-1}. Since $X_{0,l}, \ldots, X_{m-2,l}$ do not depend on W_{m-1}, we have the Markov chain $X_{0,l}, \ldots, X_{m-2,l} - \mathbf{X}_{m-1}^{l-1}\mathbf{Y}_{m-1}^{l-1} - X_{m-1,l}$. Hence, we have

$$
\begin{aligned}
P_{UX_0\ldots X_{m-1}} &= P_U P_{X_0\ldots X_{m-1}|U} \\
&= P_U P_{X_{0,l}\ldots X_{m-2,l}|l\mathbf{X}_{m-1}^{l-1}\mathbf{Y}_{m-1}^{l-1}} P_{X_{m-1,l}|l\mathbf{X}_{m-1}^{l-1}\mathbf{Y}_{m-1}^{l-1}} \\
&= P_U P_{X_0\ldots X_{m-2}|U} P_{X_{m-1}|U}.
\end{aligned}
$$

This shows $X_0, \ldots, X_{m-2} - U - X_{m-1}$. Further, the Markov chain $X_0 - \cdots - X_{m-2}$ holds since $X_{i,l}$, $0 \leq i \leq m-2$, is a function of $\mathbf{Y}_i^{l-1}$ (see (3.3)) and each $Y_{i,l}$ is a function of $X_{i-1,l}$ and $X_{i,l}$. We finally observe that the pmf $P_{X_0X_1}$ as shown in Table 3.1 can be assumed. Hence, the independence of X_0 and X_1 in (3.11) follows.

3.4.3. Cardinality Bound on U

In this section, we show that the cardinality of the range of U can be bounded by $|\mathcal{X}_{m-2}| \cdot |\mathcal{X}_{m-1}| + 2$. The proof is based on the following lemma by Ahlswede and Körner [36].

Lemma 3.2. Let $\mathcal{P}_r$ be the set of all probability r-vectors $\mathbf{p} = (p_1, \ldots, p_r)$ and let $f_j(\mathbf{p})$, $j = 1, \ldots, k$, be continuous functions on $\mathcal{P}_r$. Then, to any probability measure μ on (the Borel subsets of) $\mathcal{P}_r$ there exist $(k+1)$ elements $\mathbf{p}_i$ of $\mathcal{P}_r$ and constants $\alpha_i \geq 0$, $i = 1, \ldots, k+1$ with $\sum_{i=1}^{k+1} \alpha_i = 1$, such that

$$
\int f_j(\mathbf{p}) \, d\mu = \sum_{i=1}^{k+1} \alpha_i f_j(\mathbf{p}_i), \quad j = 1, \ldots, k.
$$

In our case, $\mathcal{P}_r$ is the set of all pmfs on $\mathcal{X}_{m-2} \times \mathcal{X}_{m-1}$. We can assume that $\mathcal{X}_{m-2} \times \mathcal{X}_{m-1}$ is ordered and we refer to the elements of $\mathcal{X}_{m-2} \times \mathcal{X}_{m-1}$ by $j = 1, 2, \ldots, r$ where $r \stackrel{def}{=} |\mathcal{X}_{m-2} \times \mathcal{X}_{m-1}|$. The conditional pmf $\left\{ P_{X_{m-2}X_{m-1}|U}(j|u) : j \in \mathcal{X}_{m-2} \times \mathcal{X}_{m-1} \right\}$ can

be interpreted as an element of $\mathcal{P}_r$ and $\{P_U(u)\}_{u \in U}$ as a Borel measure on $\mathcal{P}_r$. Let $\mathbf{p} = (p_1, \ldots, p_r) \in \mathcal{P}_r$, $\mathcal{J}_x \overset{\text{def}}{=} \{j : X_{m-1} = x\}$ and consider the following $r + 2$ continuous functions on $\mathcal{P}_r$:

▷ $f_j(\mathbf{p}) = p_j$, the projection on the jth coordinate of $\mathbf{p}$ for $j = 1, 2, \ldots, r$;

▷ $f_{r+1}(\mathbf{p}) = - \displaystyle\sum_{\substack{y \in \mathcal{Y}_{m-1} \\ x \in \mathcal{X}_{m-1}}} \left(\sum_{\mathcal{J}_x} P_{Y_{m-1}|X_{m-2}X_{m-1}}(y|j)p_j \right) \log \left(\dfrac{\sum_{\mathcal{J}_x} P_{Y_{m-1}|X_{m-2}X_{m-1}}(y|j)p_j}{\sum_{\mathcal{J}_x} p_j} \right)$;

▷ $f_{r+2}(\mathbf{p}) = - \displaystyle\sum_{x \in \mathcal{X}_{m-1}} \left(\sum_{\mathcal{J}_x} p_j \right) \log \left(\sum_{\mathcal{J}_x} p_j \right)$.

Then

$$\sum_{u \in U} f_j \left(P_{X_{m-2}X_{m-1}|U} (\cdot|u) \right) P_U(u) = P_{X_{m-2}X_{m-1}}(j)$$

for all $j = 1, 2, \ldots, r - 1$. Further,

$$\sum_{u \in U} f_{r+1} \left(P_{X_{m-2}X_{m-1}|U} (\cdot|u) \right) P_U(u) = H(Y_{m-1}|UX_{m-1})$$

$$\sum_{u \in U} f_{r+2} \left(P_{X_{m-2}X_{m-1}|U} (\cdot|u) \right) P_U(u) = H(X_{m-1}|U).$$

Note that $H(X_{m-1}|U)$ equals $H(Y_m|U)$ due to the deterministic network model. Now by Lemma 3.2, we can find a random variable U' with a range of at most $|\mathcal{X}_{m-2}| \cdot |\mathcal{X}_{m-1}| + 2$ values such that $H(Y_{m-1}|U'X_{m-1}) = H(Y_{m-1}|UX_{m-1})$ and $H(Y_m|U') = H(Y_m|U)$.

3.4.4. Remark

A closer look at the proof of the converse part in Section 3.4.2 shows that the properties of the deterministic network model were used only to bound R_{m-1}. Hence, by setting R_{m-1} equal to zero and $U = \emptyset$, we obtain the capacity of a *noisy* relay cascade, as stated next.

Theorem 3.3. Consider a discrete memoryless relay cascade composed of a source node, $m - 1$ intermediate relay nodes and a sink node as depicted in the upper part of Fig. 2.2. The links in the cascade are allowed to be noisy, i.e., they obey general pmfs $P_{Y_i|X_{i-1}X_i}$,

$i = 1, 2, \ldots, m - 1$, and $P_{Y_m|X_{m-1}}$. The capacity of the relay cascade is given by

$$C = \max_{P_{X_0 \cdots X_{m-1}}} \min \left\{ I\left(X_0; Y_1 | X_1\right), \ldots, I\left(X_{m-2}; Y_{m-1} | X_{m-1}\right), I\left(X_{m-1}; Y_m\right) \right\}.$$

Theorem 3.3 could also have been obtained by observing that the model is a physically degraded relay network and applying the decode-forward rate of Xie and Kumar [28].

3.5. Discussion

We derived the capacity region of half-duplex-constrained relay cascades with two sources. The second source was either the first or the last relay in the cascade. An obvious extension is to allow any relay in the cascade to act as a second source. This case turns out to be elusive. Though not considered in this chapter, we point out that a combination of the random code described in Section 3.4.1, tailored to the new setup, and the timing code introduced in Section 2.4 is capacity achieving.

4

A Constrained Coding Approach to Error-Free Half-Duplex Relay Networks[‡]

4.1. Introduction

Information transmission through a relay channel or network with error-free and/or half-duplex-constrained relays is a problem that has been considered by several authors [1, 2, 9, 27, 31, 33, 38, 39]. In this chapter the focus is on directed trees of error-free half-duplex-constrained relays, as shown in Fig. 2.6. Such networks include a chain of relays as a special case, which was the central topic of Chapter 2. The transmission objective is to broadcast information from a source (situated at the root of the tree) to all network nodes, each of which is half-duplex constrained. In each time slot, a node either receives

(without error) the transmission of its parent, or broadcasts information to its children, but it may not do both.

More precisely, we assume that transmission between nodes in the network occurs in discrete time-slots. Recall that $\mathcal{Q} = \{0, \ldots, q-1\}$ represents a q-ary transmission alphabet, while the additional symbol N indicates a channel use without transmission. Moreover, $\mathcal{X} = \mathcal{Q} \cup \{\mathsf{N}\}$. In any given time-slot, each node of the network broadcasts a symbol $x \in \mathcal{X}$ to its children; the node is said to be ON if $x \in \mathcal{Q}$; otherwise $x = \mathsf{N}$ and the node is said to be OFF.

In accordance with (2.2), the half-duplex constraint is captured as follows. When a relay is OFF, it is connected to its parent through a noiseless $(q+1)$-ary channel with alphabet $\mathcal{X}$, and so receives the transmission from its parent without error. When a relay is ON, it cannot receive, so the symbol sent by its parent is erased.

The simplest approach to information broadcasting is to require each network node to be OFF half of the time, organized in deterministic fashion so that a node is OFF whenever its parent might be ON. Nodes simply forward what they receive, resulting in a transmission rate of $0.5\log_2(q+1)$ bits per symbol (b/sym). By Theorem 2.2 and the explanations in Section 2.7.1, the multicast capacity, on the other hand, approaches [9]

$$C(q) = \log_2\left(\frac{1 + \sqrt{4q+1}}{2}\right) \text{ b/sym} \tag{4.1}$$

as the tree-depth becomes large.[4] In the binary case, deterministic store-and-forward achieves 0.5 b/sym whereas $C(1) = \log_2 \phi = 0.6924$ b/sym, where ϕ is the golden ratio. For trees of finite depth, even greater rates are possible. For example, for trees of depth $D = 2$, a rate of $C_1(1) = 0.7729$ b/sym is achievable in the binary case (see Table 2.5). It is clear that deterministic store-and-forward falls short of the maximum possible transmission rate. However, to achieve the multicast capacity of trees of finite depth requires a sophisticated coding approach, as introduced in Section 2.4. The approach is based on coding additional information in the ON-OFF patterns of the nodes.

We note that ON-OFF patterns have also been exploited for neighbor discovery in half-duplex-constrained networks using a compressed sensing approach [40–42]. Another

---

[4]We use the unit "b/sym" instead of "b/u" since the tools, which are applied in the following, stem from constrained coding where "b/sym" is commonly used. Moreover, $C(q)$ is used instead of $C_\infty(q)$.

problem, namely a line of three nodes where the first two nodes are half-duplex sources and where all nodes are connected by packet erasure channels, was addressed in [43] within a queuing-theoretic framework. In [44] a Gaussian point-to-point channel with a sender subject to a duty cycle constraint (e.g., a half-duplex constraint) and an average power constraint is considered. Interestingly, the optimal input distribution is shown to be discrete, i.e., a modulated ON-OFF signaling scheme is capacity-achieving.

In this chapter we will present a multicasting scheme, based on constrained coding, that preserves the simplicity of the store-and-forward approach, but achieves a higher transmission rate than deterministic store-and-forward. In particular, we show that we can achieve a multicast rate of $C(q)$ in any error-free half-duplex-constrained tree network using constrained coding at the source and symbol forwarding at the relays.

The organization of the chapter is as follows. Section 4.2 provides a brief summary of constrained coding. The multicasting scheme is introduced in Section 4.3. In Section 4.4, we compare rates achievable in trees of finite depth using the timing code of Chapter 2 and the code proposed in this chapter. The chapter is ended with an appendix on the state-splitting algorithm and its usage to construct a $3/2$ finite-state encoder.

4.2. Constrained Coding Background

The approach we take to multicasting in a tree of half-duplex-constrained nodes uses tools from constrained coding (or symbolic dynamics); see, e.g., [45, 46]. In a nutshell, the field of constrained coding studies mappings from unconstrained input sequences to output sequences obeying certain constraints. The constraints are often expressed by specifying forbidden sub-blocks, i.e., subsequences that are not permitted to occur in any output sequence. A classical example is the *golden mean shift*, which is the set of binary sequences in which the sub-block 11 never occurs. Constrained coding has found many applications in magnetic and optical recording systems.

The *capacity* of a constrained system, which is the maximum rate at which unconstrained binary data may be mapped to constrained output data, is defined as

$$C = \limsup_{n \to \infty} \frac{1}{n} \log_2 N(n) \text{ b/sym,}$$

where $N(n)$ denotes the number of sequences in the output alphabet having length n and satisfying the given constraint. For example, the golden mean shift satisfies the *Fibonacci recurrence*: for $n \geq 2$,

$$N(n) = N(n-1) + N(n-2), \text{ with } N(0) = 1, \ N(1) = 2.$$

From this it can be shown that the golden mean shift has $C = \log_2 \phi$, where $\phi = (1 + \sqrt{5})/2$ is the golden ratio (a result that explains the name "golden mean shift"). Interestingly, the golden ratio also arises in the analysis of the trapdoor channel [7] with feedback, where it is shown that the capacity equals $\log_2 \phi$.

It is well known that the capacity of certain constrained systems can be obtained via an irreducible, lossless graph presentation of the constraint [45]. If G is such a presentation, and A_G is the adjacency matrix of G, then

$$C = \log_2 \lambda(A_G),$$

where λ is the largest of the absolute values of the eigenvalues of A_G. This formulation of capacity will be used in the sequel.

4.3. Code Construction

We now describe the constrained coding approach taken in this paper. The transmission protocol is trivial, amounting to simple symbol-forwarding: during any given time-slot, every non-source node simply forwards (to all of its children) the symbol it has received from its parent during the previous time-slot. Correct forwarding is achieved provided that nodes obey the half-duplex constraint, i.e., that they are never ON when their parent node might be ON. Under the symbol-forwarding protocol, this is accomplished if and only if the source is never itself ON in two adjacent time-slots.

Thus we arrive naturally at a constrained coding problem: the source may emit any sequence of symbols drawn from $\mathcal{X}$ satisfying the constraint that no two adjacent symbols are drawn from $\mathcal{Q}$. In the language of symbolic dynamics, every transmitted sequence is drawn from the shift of finite type denoted as $\mathcal{X}_{\mathcal{Q}^2}$ having forbidden sub-block set

$Q^2 \stackrel{def}{=} Q \times Q$. An irreducible, lossless graph presentation of this shift is shown in Fig. 4.1. When $q = 1$, this shift is equivalent to the golden mean shift.
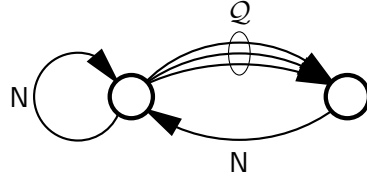
Figure 4.1.: Graph presentation of half-duplex constraint under symbol-forwarding.

The adjacency matrix of this presentation is given, as a function of q, as

$$A(q) = \begin{bmatrix} 1 & q \\ 1 & 0 \end{bmatrix},$$

which has characteristic polynomial

$$p_q(\lambda) = \lambda^2 - \lambda - q.$$

The eigenvalues of $A(q)$ (the roots of $p_q(\lambda)$) are given as

$$\lambda = \frac{1}{2}\left(1 \pm \sqrt{1 + 4q}\right),$$

and the constrained capacity (the logarithm of the largest eigenvalue) is given as

$$C(q) = \log_2\left(\frac{1 + \sqrt{1 + 4q}}{2}\right).$$

Remarkably—and this is the central result of this chapter—the constrained coding approach achieves the multicast capacity $C(q)$ of infinite-depth trees, but without the necessity of designing sophisticated timing codes as we did in Chapter 2 (see also [9]).

The capacity $C(q)$ can be approached using methods (e.g., the state-splitting algorithm) from constrained coding. Fig. 4.2 provides two examples. The first, in Fig. 4.2(a), is a standard example in constrained coding [46] and gives a rate-(2/3) encoder for $q = 1$, which achieves more than 96% of the capacity $C(1) = \log_2 \phi$. The second, in
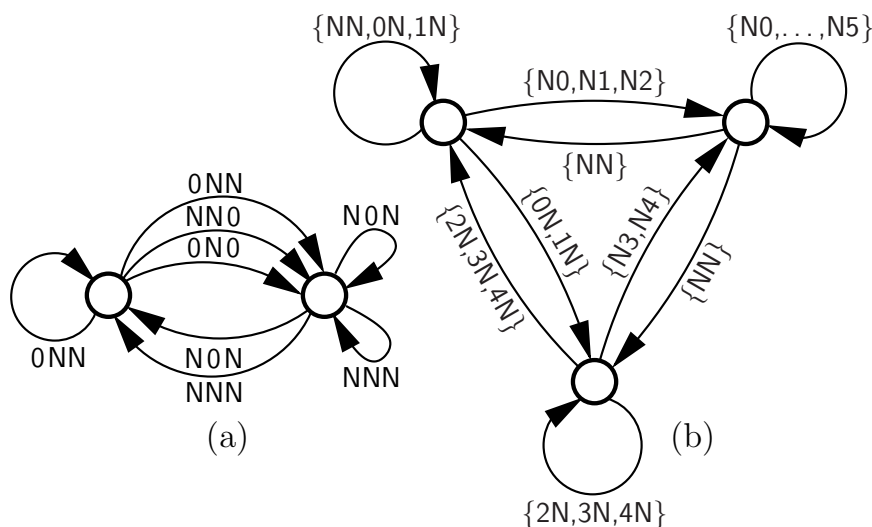
Figure 4.2.: Encoders for (a) $q = 1$, $R = 2/3$, $C(1) = \log_2 \phi \approx 0.6942$ b/sym, (b) $q = 6$, $R = 3/2$, $C(6) = \log_2 3 \approx 1.5850$ b/sym.

Fig. 4.2(b), is a rate-$(3/2)$ encoder for $q = 6$, which achieves more than 94% of the capacity $C(6) = \log_2(3)$. Similar examples can readily be constructed for other values of q. For any given q, if the number of encoder states is allowed to grow, $C(q)$ can be approached arbitrarily closely.

4.4. Discussion

Table 5.1 compares, for $q = 1$, rates achievable in networks of finite depth D using three approaches: the timing codes presented in Chapter 2, the constrained coding approach of this chapter, and the deterministic store-and-forward approach. The row labeled C_{D-1}, which gives the maximum achievable rate (using timing codes), serves as a benchmark for the other schemes (see also Table 2.5 for the values of C_{D-1}). We observe that $C(q)/C_{D-1}$, the relative efficiency of constrained coding, rapidly converges to unity as D increases. On the other hand, $0.5/C_{D-1}$, the relative efficiency of deterministic store-and-forward, saturates at approximately 72%. The differences among the three approaches however become smaller as q increases.

We observe that timing codes require nodes at different depths in a finite tree to have different (carefully designed) ON-OFF duty cycles that depend on the depth of the

Table 4.1.: Achievable Rates in Networks of Finite Depth D with $q = 1$

D	2	3	5	11	∞
C_{D-1} [b/sym]	0.7729	0.7324	0.7099	0.6981	0.6942
$C(q)/C_{D-1}$ [%]	89.82	94.79	97.80	99.44	100
$0.5/C_{D-1}$ [%]	64.70	68.27	70.43	71.62	72.02

tree. It is shown in Table 2.4 that these duty cycles converge to a constant as the tree depth grows. The capacity $C(q)$, achieved both by timing codes and by the constrained coding approach of this paper, is the maximum rate that can be achieved with a constant ON-OFF duty cycle throughout the network.

We note that it might be interesting to consider a network model with noisy transmission links. In this case, techniques that combine constrained coding with error control (as in, e.g., [47–49]) may be helpful. As a final remark, we point out that encoders, which output sequences with a minimum number of ON- and OFF-symbols in a row, would greatly relieve the requirements on synchronization and switching speed of the relays. Such encoders can be easily constructed along the lines outlined in the following appendix.

4.A. Appendix

In this appendix, we show how to construct the rate 3/2 finite-state encoder depicted in Fig. 4.2(b) using the *state-splitting algorithm*. The definition of a finite-state encoder is as follows.

Definition 4.1. [46, Chapter 4] Let S be a constrained system (like the one shown in Fig. 4.1). A *rate $p : q$ finite-state encoder* for S is a labeled graph H such that

  ▷ each state of H has out-degree 2^p;

  ▷ each edge of H is labeled with q symbols from the code alphabet;

  ▷ $S(H) \subseteq S$, i.e, the set of possible sequences resulting from parsing the edges of H is contained in S;

▷ H is lossless, i.e., any two distinct paths with the same initial state and terminal state have different labelings.

The following result, known as *Finite-State Coding Theorem*, reveals that the capacity C of a constrained system can be approximated arbitrarily close by finite-state encoders.

Theorem 4.1. [46, Chapter 4] Let S be a constrained system. If $p/q \leq C$, then there exists a rate $p : q$ finite-state encoder for S.

The proof of Theorem 4.1 gives a procedure—the state-splitting algorithm—to construct rate $p : q$ finite-state encoders. In a nutshell, for a given deterministic presentation G of a constrained system S (which is not already a finite-state encoder according to Definition 4.1), the state-splitting algorithm applies iteratively a sequence of state-splitting transformations beginning with the qth power graph G^q. The procedure yields a new graphical presentation of the constrained system with minimum out-degree at least 2^p; then, after deleting edges such that every node has out-degree 2^p, we get a rate $p : q$ finite-state encoder for S.

The rules for a single state-splitting transformation are described next [46, Chapter 5]. Let $H = (\mathcal{V}, \mathcal{E})$ be a labeled graph with a finite set of states $\mathcal{V}$ and a finite set of edges $\mathcal{E}$. Let $\mathcal{E}_u$ denote the set of outgoing edges from state u in H. A splitting of state $u \in \mathcal{V}$ into two descendant states u_1 and u_2 is determined by a partition of $\mathcal{E}_u$ into two disjoint sets $\mathcal{E}_{u_1}$ and $\mathcal{E}_{u_2}$. The new labeled graph $H' = (\mathcal{V}', \mathcal{E}')$, which has state set $\mathcal{V}' = V \setminus \{u\} \cup \{u_1, u_2\}$, describes the same constrained system as H if the set of edges $\mathcal{E}'$ are chosen according to the following rules:

▷ The edges in H' that do not involve states u_1 and u_2 are inherited from H and have the same labeling;

▷ Let edge e in H start at a state $v \neq u$ and terminate in state u. This edge is replicated in H' to produce two edges (each with the same labeling as the original edge in H): an edge from v to u_1 and an edge from v to u_2;

▷ Let edge e in H start at state u and terminate in a state $v \neq u$, and suppose e belongs to the set $\mathcal{E}_{u_i}$, $i \in \{1, 2\}$, in the partition of $\mathcal{E}_u$. We draw in H' a corresponding edge from state u_i to state v with the same labeling as in H;

▷ Let edge e be a self-loop at state u in H, and suppose that e belongs to $\mathcal{E}_{u_i}$, $i \in \{1, 2\}$. In H' there will be two edges from state u_i corresponding to e (each with the same labeling as the original self-loop in H): one edge to state u_1, the other to state u_2.

We are now ready to construct the encoder of Fig. 4.2b. This is done in three steps as outlined in Fig. 4.3. First, the second power graph G^2, depicted in Fig. 4.3a, is generated from the golden mean shift shown in Fig 4.1 where $\mathcal{Q} = \{0, \dots, 5\}$. Subsequently, the left state of G^2 (the state with more outgoing edges) is splitted (according to the splitting criterions of the state-splitting algorithm), resulting in the graph shown in Fig. 4.3b. The splitting is based on the following partition of the outgoing edges of the splitted state: $\{\mathsf{NN}, \mathsf{0N}, \mathsf{1N}, \mathsf{N0}, \mathsf{N1}, \mathsf{N2}\}$ and $\{\mathsf{2N}, \mathsf{3N}, \mathsf{4N}, \mathsf{5N}, \mathsf{N3}, \mathsf{N4}, \mathsf{N5}\}$. The first set corresponds to the upper left state in Fig. 4.3b while the second set is with respect to the bottom state. In the final step, as many edges from Fig. 4.3b are deleted such that eight outgoing edges per state remain (see Fig. 4.3c). Observe that the deletion of edges is the reason why the rate of the resulting $3 : 2$ encoder is less than the capacity $C(6) = \log_2 3$ b/sym of the constrained system.
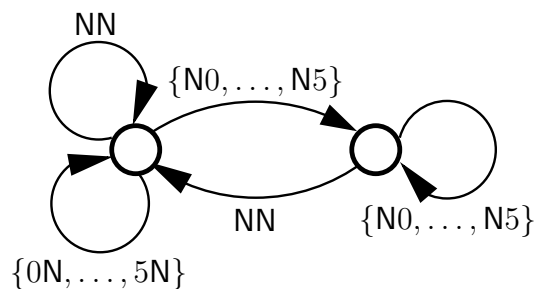
(a) The second power graph G^2 of the graph shown in Fig 4.1 where $\mathcal{Q} = \{0, \dots, 5\}$.

{NN,0N,1N}   {N0,...,N5}

{N0,N1,N2}

{NN}

{NN,0N,1N}

{N3,N4,N5}

{2N,3N,4N,5N}

{NN}

{2N,3N,4N,5N}
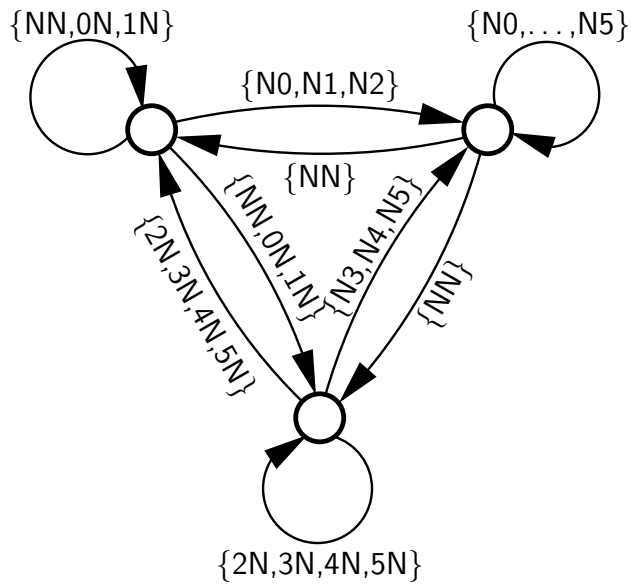
(b) Graph resulting from splitting the left state of G^2.

{NN,0N,1N}   {N0,...,N5}

{N0,N1,N2}

{NN}

{NN,0N,1N}

{N3,N4,N5}

{2N,3N,4N,5N}

{NN}

{2N,3N,4N,5N}
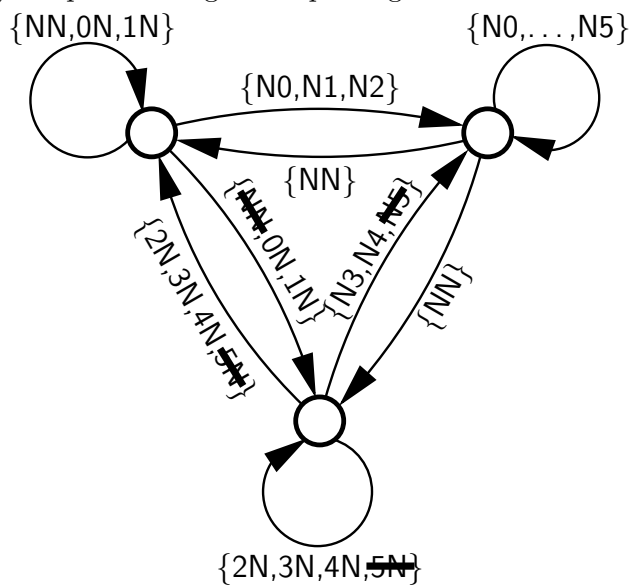
(c) Deletion of edges such that 8 outgoing edges per state remain.

Figure 4.3.: Construction of the rate 3/2 finite-state encoder depicted in Fig. 4.2(b).

5

Recursions for the Trapdoor Channel and an Upper Bound on its Capacity[§]

5.1. Introduction and Channel Model

The focus of this chapter is on the trapdoor channel. The trapdoor channel was introduced by David Blackwell in 1961 [5] and is used by Robert Ash both as a book cover and as an introductory example for channels with memory [52]. The mapping of channel inputs to channel outputs can be described as follows. Consider a box that contains a ball that is labeled $s_0 \in \{0, 1\}$, where the index 0 refers to time 0. Both the sender and the receiver know the initial ball. In time slot 1, the sender places a new ball labeled $x_1 \in \{0, 1\}$ in the box. In the same time slot, the receiver chooses one of the two balls s_0

[§]This chapter is based on the following publication [50]: *Proc. IEEE Int. Symp. Inf. Theory,* Honolulu, HI, USA, Jun. 29 - Jul. 4, 2014, pp. 2914–2918, submitted to *IEEE Trans. Inf. Theory* [51].

(a) The trapdoor channel at time t.

(b) The trapdoor channel at time $t + 1$. Here $y_t = s_{t-1}$.
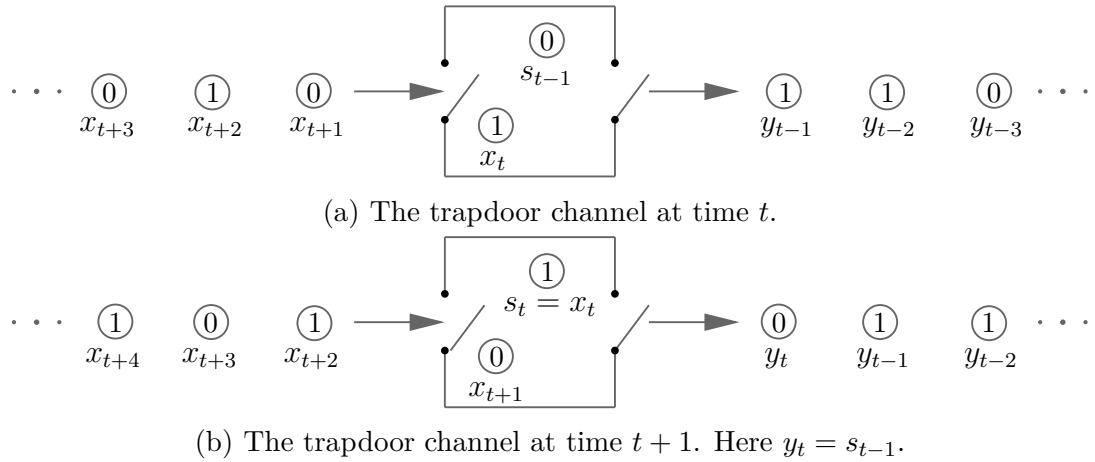
Figure 5.1.: The trapdoor channel.

or x_1 at random while the other ball remains in the box. The chosen ball is interpreted as channel output y_1 at time $t = 1$ while the remaining ball becomes the channel state s_1. The same procedure is applied in every future channel use. In time slot 2, for instance, the sender places a new ball $x_2 \in \{0, 1\}$ in the box and the corresponding channel output y_2 is either x_2 or s_1. The transmission process is visualized in Fig. 5.1. Fig. 5.1a shows the trapdoor channel at time t when the sender places ball x_t in the box. In the same time slot, the receiver chooses randomly one of the two balls x_t or s_{t-1} as channel output, in the figure the ball labeled with s_{t-1}. Consequently, the upcoming channel state s_t becomes x_t (see Fig. 5.1b). At time $t + 1$ the sender places a new ball x_{t+1} in the box and the receiver draws y_{t+1} from s_t and x_{t+1}. Table 5.1 depicts the probability of an output y_t given an input x_t and state s_{t-1}.

Despite the simplicity of the trapdoor channel, deriving its capacity seems challenging and is an open problem. One feature that makes the problem cumbersome is that the distribution of the output symbols may depend on events happening arbitrarily far back in the past since each ball has a positive probability to remain in the channel over any finite number of channel uses. Instead of maximizing $I(X; Y)$ one rather has to consider the multi-letter mutual information, i.e., $\limsup_{n \to \infty} I(\mathbf{X}^n; \mathbf{Y}^n)$.

Let $\mathbf{P}_{n|s_0}$ denote the matrix of conditional probabilities of output sequences of length n given input sequences of length n where the initial state equals s_0. The following ordering of the entries of $\mathbf{P}_{n|s_0}$ is assumed. Row indices represent input sequences and column

Table 5.1.: Transition Probabilities of the Trapdoor Channel

x_t	s_{t-1}	$P_{Y_t|X_t S_{t-1}}(y_t = 0|x_t, s_{t-1})$	$P_{Y_t|X_t S_{t-1}}(y_t = 1|x_t, s_{t-1})$
0	0	1	0
0	1	0.5	0.5
1	0	0.5	0.5
1	1	0	1

indices represent output sequences. To be more precise, the $(i,j)^{\text{th}}$ entry of $\left[\mathbf{P}_{n|s_0}\right]$, indicated as $\left[\mathbf{P}_{n|s_0}\right]_{i,j}$, is the conditional probability of the binary output sequence corresponding to the integer $j-1$ given the binary input sequence corresponding the the integer $i-1$, $1 \le i, j \le 2^n$. For instance, if $n = 3$, then $\left[\mathbf{P}_{3|s_0}\right]_{5,3}$ denotes the conditional probability that the channel input $x_1, x_2, x_3 = 1, 0, 0$ will be mapped to the channel output $y_1, y_2, y_3 = 0, 1, 0$. It was shown in [53] that $\mathbf{P}_{n|s_0}$, $s_0 \in \{0, 1\}$, satisfies the recursion laws

$$\mathbf{P}_{n+1|0} = \begin{bmatrix} \mathbf{P}_{n|0} & \mathbf{0} \\ \frac{1}{2}\mathbf{P}_{n|1} & \frac{1}{2}\mathbf{P}_{n|0} \end{bmatrix} \tag{5.1}$$

$$\mathbf{P}_{n+1|1} = \begin{bmatrix} \frac{1}{2}\mathbf{P}_{n|1} & \frac{1}{2}\mathbf{P}_{n|0} \\ \mathbf{0} & \mathbf{P}_{n|1} \end{bmatrix}, \tag{5.2}$$

where the initial matrices are given by $\mathbf{P}_{0|0} = \mathbf{P}_{0|1} = 1$. Ahlswede and Kaspi [54] derived the *zero-error capacity* of the trapdoor channel, which equals 0.5 b/u. Permuter et al. [7] considered the trapdoor channel under the additional assumption of having a unit delay feedback link available from the receiver to the sender. They established that the capacity of the trapdoor channel with feedback is equal to the logarithm of the golden ratio. The achievability scheme involves a constrained coding scheme, similar to the one introduced in Chapter 4.

In this chapter, we consider the problem of maximizing the n-letter mutual information of the trapdoor channel for any $n \in \mathbb{N}$. We relax the problem by permitting distributions that are not probability distributions. The resulting optimization problem is convex but the feasible set is larger than the probability simplex. Using the method of Lagrange multipliers via a theorem presented in [52], we find explicit solutions for any $n \in \mathbb{N}$. It

is then shown that $\frac{1}{2}\log_2\left(\frac{5}{2}\right) \approx 0.6610$ b/u is an upper bound on the capacity of the trapdoor channel. Specifically, the same absolute maximum $\frac{1}{2}\log_2\left(\frac{5}{2}\right) \approx 0.6610$ b/u results for all trapdoor channels which process input blocks of even length. And the sequence of absolute maxima corresponding to trapdoor channels which process inputs of odd lengths converges to $\frac{1}{2}\log_2\left(\frac{5}{2}\right)$ b/u from below as the block length increases. Unfortunately, the absolute maxima of our relaxed optimization are attained outside the probability simplex. Otherwise we would have established the capacity. Nevertheless, $\frac{1}{2}\log_2\left(\frac{5}{2}\right) \approx 0.6610$ b/u is, to the best of our knowledge, the tightest capacity upper bound. Moreover, this bound is less than the feedback capacity of the trapdoor channel proving that feedback increases the capacity.

The organization of the chapter is as follows. Section 5.2 presents the derivation of the upper bound. In particular, the problem is set up and a useful result from the literature is reviewed. Two recursions are then developed for the trapdoor channel based on which the main result is derived. The chapter is concluded with Section 5.3.

5.2. A Lagrange Multiplier Approach to the Trapdoor Channel

5.2.1. Problem Formulation

We derive an upper bound on the capacity of the trapdoor channel. Specifically, for any $n \in \mathbb{N}$, we find a solution to the optimization problem

$$\underset{P_{\mathbf{X}^n}}{\text{maximize}} \quad \frac{1}{n} I\left(\mathbf{X}^n; \mathbf{Y}^n | s_0\right)$$

$$= \frac{1}{n} \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} p_i \left[\mathbf{P}_{n|s_0}\right]_{i,j} \log \frac{\left[\mathbf{P}_{n|s_0}\right]_{i,j}}{\sum_{k=1}^{2^n} p_k \left[\mathbf{P}_{n|s_0}\right]_{k,j}} \tag{5.3}$$

$$\text{subject to} \quad \sum_{i=1}^{2^n} p_i = 1 \tag{5.4}$$

$$\sum_{k=1}^{2^n} p_k \left[\mathbf{P}_{n|s_0}\right]_{k,j} \geq 0 \quad \text{for all } 1 \leq j \leq 2^n. \tag{5.5}$$

Note that $P_{\mathbf{X}^n}$ is a 2^n-sequences $(p_1, \ldots, p_{2^n})$ where p_i denotes the probability of the ith input sequence $\mathbf{x}_i$, i.e., the binary sequence corresponding to the integer $i-1$. Constraint (5.5) guarantees that the argument of the logarithm does not become negative. The feasible set, defined by (5.4) and (5.5), is convex. It includes the set of probability mass functions, but might be larger. To see this note that (5.5) is a weighted sum of all p_k where each weight $\left[\mathbf{P}_{n|s_0}\right]_{k,j}$ is non negative. Clearly, (5.4) and (5.5) are satisfied by probability distributions. However, there might exist "distributions" which involve negative values and sum up to one but still satisfy (5.5). Moreover, the objective function $n^{-1}I\left(\mathbf{X}^n; \mathbf{Y}^n | s_0\right)$ is concave on the set of "distributions" satisfying (5.4) and (5.5). Consequently, the optimization problem is convex and every solution maximizes $n^{-1}I\left(\mathbf{X}^n; \mathbf{Y}^n | s_0\right)$. In the following, we use the notation

$$C_n^{\uparrow} \stackrel{\text{def}}{=} \max_{P_{\mathbf{X}^n}} n^{-1}I\left(\mathbf{X}^n; \mathbf{Y}^n | s_0\right).$$

Taking the limit of the sequence $\left(C_n^{\uparrow}\right)_{n \in \mathbb{N}}$, one obtains either the capacity of the trapdoor channel or an upper bound on the capacity, depending on whether the limit is attained inside or outside the set of probability distributions. Since it does not matter whether the optimization is with respect to initial state 0 or 1 (due to symmetry reasons), we do not have to distinguish between *lower capacity* and *upper capacity* [6, Chapter 4.6]

5.2.2. Using a Result from the Literature

The reason for considering (5.5) and not the more natural constraints $p_k \geq 0$ for all k is that a closed form solution can be obtained by applying the method of *Lagrange multipliers* to (5.3) and (5.4). As a byproduct, (5.5) will be automatically satisfied. In particular, setting the partial derivatives of

$$\frac{1}{n}I\left(\mathbf{X}^n; \mathbf{Y}^n | s_0\right) + \lambda \sum_{i=1}^{2^n} p_i \tag{5.6}$$

with respect to each of the p_i equal to zero results in a closed form solution of the considered optimization problem.

This was done in [52, Theorem 3.3.3] for general discrete memoryless channels which

are square and non singular. Note that $\mathbf{P}_{n|s_0}$ is square and non singular (see, e.g., Lemma 5.1 (b)). Moreover, we assume that the channel $\mathbf{P}_{n|s_0}$ is memoryless by repeatedly using it over a large number of input blocks of length n. Consequently, $C_n^{\uparrow}$ might be an upper bound on the capacity of the channel $\mathbf{P}_{n|s_0}$. The reason is that some input blocks possibly drive the channel $\mathbf{P}_{n|s_0}$ into the opposite state $s_0 \oplus 1$, i.e., the upcoming input block sees channel $\mathbf{P}_{n|s_0 \oplus 1}$ (whose $C_n^{\uparrow}$ is equal to $C_n^{\uparrow}$ of $\mathbf{P}_{n|s_0}$ by symmetry) but not $\mathbf{P}_{n|s_0}$. However, by assuming that the channel does not change over time, the sender always knows the channel state before a new block is transmitted. Hence, $C_n^{\uparrow}$ might be an upper bound (even though it is attained on the set of probability distributions). Nevertheless, this issue can be ignored if n goes to infinity because in the asymptotic regime the channel $\mathbf{P}_{n|s_0}$ is used only once.

In summary, it is valid to apply [52, Theorem 3.3.3] which yields

$$C_n^{\uparrow} = \frac{1}{n} \log_2 \sum_{j=1}^{2^n} \exp_2 \left(-\sum_{i=1}^{2^n} \left[\mathbf{P}_{n|s_0}^{-1} \right]_{j,i} H\left(\mathbf{Y}^n | \mathbf{X}^n = \mathbf{x}_i \right) \right), \tag{5.7}$$

attained at

$$p_k = 2^{-C_n^{\uparrow}} d_k, \quad k = 1, \ldots, 2^n \tag{5.8}$$

where

$$d_k = \sum_{j=1}^{2^n} \left[\mathbf{P}_{n|s_0}^{-1} \right]_{j,k} \exp_2 \left(-\sum_{i=1}^{2^n} \left[\mathbf{P}_{n|s_0}^{-1} \right]_{j,i} H\left(\mathbf{Y}^n | \mathbf{X}^n = \mathbf{x}_i \right) \right). \tag{5.9}$$

Clearly, $(p_1, \ldots, p_{2^n})$ is a probability distribution only if $d_k \geq 0$. The Lagrangian (5.6) does not involve constraint (5.5). However, the proof of [52, Theorem 3.3.3] shows that

$$\sum_{k=1}^{2^n} p_k \left[\mathbf{P}_{n|s_0} \right]_{k,j} = \exp \left(\lambda - \sum_{i=1}^{M} \left[\mathbf{P}_{n|s_0}^{-1} \right]_{j,i} H\left(\mathbf{Y}^n | \mathbf{X}^n = \mathbf{x}_i \right) - 1 \right) \tag{5.10}$$

for all $1 \leq j \leq 2^n$. Hence, (5.5) is satisfied.

For computational reasons, we write (5.7) in matrix vector notation, which reads

$$C_n^{\uparrow} = \frac{1}{n} \log_2 \left\{ \mathbf{1}_n^T \exp_2 \left[\mathbf{P}_{n|s_0}^{-1} \left(\mathbf{P}_{n|s_0} \circ \log_2 \mathbf{P}_{n|s_0} \right) \mathbf{1}_n \right] \right\}, \tag{5.11}$$

where $\mathbf{1}_n$ is a column vector of length 2^n consisting only of ones while $\circ$ denotes the

Hadamard product. Observe that

$$
- \left(\mathbf{P}_{n|s_0} \circ \log_2 \mathbf{P}_{n|s_0} \right) \mathbf{1}_n = \left[H \left(\mathbf{Y}^n | \mathbf{X}^n = \mathbf{x}_1 \right), \dots, H \left(\mathbf{Y}^n | \mathbf{X}^n = \mathbf{x}_{2^n} \right) \right]^T. \tag{5.12}
$$

In the remainder, we use (5.11) instead of (5.7). In particular, we find exact numerical expressions for (5.11) in Theorem 5.1 below.

5.2.3. Useful Recursions

Definition 5.1. (a) The *conditional entropy vector* $\mathbf{h}_{n|s_0}$ of $\mathbf{P}_{n|s_0}$, $s_0 \in \{0, 1\}$, is

$$
\mathbf{h}_{n|s_0} \overset{def}{=} \left[H(\mathbf{Y}^n | \mathbf{X}^n = \mathbf{x}_1) \quad \dots \quad H(\mathbf{Y}^n | \mathbf{X}^n = \mathbf{x}_{2^n}) \right]^T \tag{5.13}
$$

$$
= - \left(\mathbf{P}_{n|s_0} \circ \log_2 \mathbf{P}_{n|s_0} \right) \mathbf{1}_n, \tag{5.14}
$$

where $n \in \mathbb{N}_0$.

(b) The *weighted conditional entropy vector* $\boldsymbol{\omega}_{n|s_0}$ of $\mathbf{P}_{n|s_0}$, $s_0 \in \{0, 1\}$, is

$$
\boldsymbol{\omega}_{n|s_0} \overset{def}{=} -\mathbf{P}_{n|s_0}^{-1} \cdot \mathbf{h}_{n|s_0} \tag{5.15}
$$

$$
= \mathbf{P}_{n|s_0}^{-1} \left(\mathbf{P}_{n|s_0} \circ \log_2 \mathbf{P}_{n|s_0} \right) \mathbf{1}_n, \tag{5.16}
$$

where $n \in \mathbb{N}_0$.

The following three lemmas provide tools that we need in order to prove recursions for $\mathbf{h}_{n|s_0}$ and $\boldsymbol{\omega}_{n|s_0}$, as stated in Lemma 5.4 and Lemma 5.5.

Lemma 5.1. (a) The trapdoor channel matrices $\mathbf{P}_{2n+2|0}$ and $\mathbf{P}_{2n+2|1}$, $n \in \mathbb{N}_0$, satisfy the following recursions:

$$
\mathbf{P}_{2n+2|0} = \begin{bmatrix} \mathbf{P}_{2n|0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \frac{1}{2}\mathbf{P}_{2n|1} & \frac{1}{2}\mathbf{P}_{2n|0} & \mathbf{0} & \mathbf{0} \\ \frac{1}{4}\mathbf{P}_{2n|1} & \frac{1}{4}\mathbf{P}_{2n|0} & \frac{1}{2}\mathbf{P}_{2n|0} & \mathbf{0} \\ \mathbf{0} & \frac{1}{2}\mathbf{P}_{2n|1} & \frac{1}{4}\mathbf{P}_{2n|1} & \frac{1}{4}\mathbf{P}_{2n|0} \end{bmatrix} \tag{5.17}
$$

$$\mathbf{P}_{2n+2|1} = \begin{bmatrix} \frac{1}{4}\mathbf{P}_{2n|1} & \frac{1}{4}\mathbf{P}_{2n|0} & \frac{1}{2}\mathbf{P}_{2n|0} & \mathbf{0} \\ \mathbf{0} & \frac{1}{2}\mathbf{P}_{2n|1} & \frac{1}{4}\mathbf{P}_{2n|1} & \frac{1}{4}\mathbf{P}_{2n|0} \\ \mathbf{0} & \mathbf{0} & \frac{1}{2}\mathbf{P}_{2n|1} & \frac{1}{2}\mathbf{P}_{2n|0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_{2n|1} \end{bmatrix}. \tag{5.18}$$

(b) Let $\mathbf{M}_0 \stackrel{def}{=} \mathbf{P}_{2n|0}^{-1}\mathbf{P}_{2n|1}\mathbf{P}_{2n|0}^{-1}$ and $\mathbf{M}_1 \stackrel{def}{=} \mathbf{P}_{2n|1}^{-1}\mathbf{P}_{2n|0}\mathbf{P}_{2n|1}^{-1}$. The inverses of $\mathbf{P}_{2n+2|0}$ and $\mathbf{P}_{2n+2|1}$, $n \in \mathbb{N}_0$, satisfy the following recursions:

$$\mathbf{P}_{2n+2|0}^{-1} = \begin{bmatrix} \mathbf{P}_{2n|0}^{-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{M}_0 & 2\mathbf{P}_{2n|0}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{P}_{2n|0}^{-1} & 2\mathbf{P}_{2n|0}^{-1} & \mathbf{0} \\ 2\mathbf{M}_0\mathbf{P}_{2n|1}\mathbf{P}_{2n|0}^{-1} & -3\mathbf{M}_0 & -2\mathbf{M}_0 & 4\mathbf{P}_{2n|0}^{-1} \end{bmatrix} \tag{5.19}$$

$$\mathbf{P}_{2n+2|1}^{-1} = \begin{bmatrix} 4\mathbf{P}_{2n|1}^{-1} & -2\mathbf{M}_1 & -3\mathbf{M}_1 & 2\mathbf{M}_1\mathbf{P}_{2n|0}\mathbf{P}_{2n|1}^{-1} \\ \mathbf{0} & 2\mathbf{P}_{2n|1}^{-1} & -\mathbf{P}_{2n|1}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 2\mathbf{P}_{2n|1}^{-1} & -\mathbf{M}_1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_{2n|1}^{-1} \end{bmatrix}. \tag{5.20}$$

Proof. (a): Substituting $\mathbf{P}_{2n+1|0}$ and $\mathbf{P}_{2n+1|1}$ into $\mathbf{P}_{2n+2|0}$ and $\mathbf{P}_{2n+2|1}$, where the four matrices are expressed as in (5.1) and (5.2), yields (5.17) and (5.18).

(b): Two versions of the matrix inversion lemma are [55]

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ -\mathbf{D}^{-1}\mathbf{C}\mathbf{A}^{-1} & \mathbf{D}^{-1} \end{bmatrix} \tag{5.21}$$

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}\mathbf{D}^{-1} \\ \mathbf{0} & \mathbf{D}^{-1} \end{bmatrix}. \tag{5.22}$$

Now divide (5.17) and (5.18) into four blocks of equal size. A twofold application of (5.21) and (5.22), first to $\mathbf{P}_{2n+2|0}$ and $\mathbf{P}_{2n+2|1}$ and, subsequently, to each of the blocks of $\mathbf{P}_{2n+2|0}$

and $\mathbf{P}_{2n+2|1}$ yields (5.19) and (5.20). $\qquad\qquad\blacksquare$

Lemma 5.2. Let $\tilde{\mathbf{I}}_n$ be the $2^n \times 2^n$ matrix whose secondary diagonal entries are equal to 1 while the remaining entries are 0. Let $\mathbf{A}$ be an arbitrary $2^n \times 2^n$ matrix and $\mathbf{b}$ an arbitrary column vector of size 2^n. A left and right multiplication of $\mathbf{A}$ with $\tilde{\mathbf{I}}_n$ results in a permutation of the elements of $\mathbf{A}$. In particular, the element $[\mathbf{A}]_{i,j}$ of $\mathbf{A}$ is shifted to position $(2^n + 1 - i, 2^n + 1 - j)$ in $\tilde{\mathbf{I}}_n \mathbf{A} \tilde{\mathbf{I}}_n$, $1 \le i, j \le 2^n$. Similarly, a left multiplication of $\mathbf{b}$ with $\tilde{\mathbf{I}}_n$ turns $\mathbf{b}$ upside down, i.e., the ith entry of $\mathbf{b}$ is shifted to the $(2^n + 1 - i)^{\text{th}}$ position in $\tilde{\mathbf{I}}_n \mathbf{b}$, $1 \le i \le 2^n$. Moreover, $\left(\tilde{\mathbf{I}}_n \mathbf{A} \tilde{\mathbf{I}}_n\right) \circ \log_2\left(\tilde{\mathbf{I}}_n \mathbf{A} \tilde{\mathbf{I}}_n\right) = \tilde{\mathbf{I}}_n \left(\mathbf{A} \circ \log_2 \mathbf{A}\right) \tilde{\mathbf{I}}_n$.

Proof. The first two properties follow from the rules of matrix multiplication and noting that the ith row and the ith column of $\tilde{\mathbf{I}}_n$ has a one at position $2^n + 1 - i$ and zeros else. The final equality holds because it does not matter whether the Hadamard product and the elementwise logarithm is applied before or after permuting the elements of $\mathbf{A}$. $\qquad\blacksquare$

A transformation relating $\mathbf{P}_{n|0}$ to $\mathbf{P}_{n|1}$, $\mathbf{P}_{n|0}^{-1}$ to $\mathbf{P}_{n|1}^{-1}$, $\mathbf{h}_{n|0}$ to $\mathbf{h}_{n|1}$ and $\boldsymbol{\omega}_{n|0}$ to $\boldsymbol{\omega}_{n|1}$ is derived next.

Lemma 5.3. Let $\mathbf{P}_{n|0}$ and $\mathbf{P}_{n|1}$ be trapdoor channel matrices, $n \in \mathbb{N}_0$. We have the following identities:

(a)

$$\mathbf{P}_{n|1} = \tilde{\mathbf{I}}_n \mathbf{P}_{n|0} \tilde{\mathbf{I}}_n \tag{5.23}$$

$$\mathbf{P}_{n|0} = \tilde{\mathbf{I}}_n \mathbf{P}_{n|1} \tilde{\mathbf{I}}_n \tag{5.24}$$

(b)

$$\mathbf{P}_{n|1}^{-1} = \tilde{\mathbf{I}}_n \mathbf{P}_{n|0}^{-1} \tilde{\mathbf{I}}_n \tag{5.25}$$

$$\mathbf{P}_{n|0}^{-1} = \tilde{\mathbf{I}}_n \mathbf{P}_{n|1}^{-1} \tilde{\mathbf{I}}_n \tag{5.26}$$

(c)

$$\mathbf{h}_{n|1} = \tilde{\mathbf{I}}_n \mathbf{h}_{n|0} \tag{5.27}$$

$$\mathbf{h}_{n|0} = \tilde{\mathbf{I}}_n \mathbf{h}_{n|1} \tag{5.28}$$

(d)

$$\boldsymbol{\omega}_{n|1} = \tilde{\mathbf{I}}_n \boldsymbol{\omega}_{n|0} \tag{5.29}$$

$$\boldsymbol{\omega}_{n|0} = \tilde{\mathbf{I}}_n \boldsymbol{\omega}_{n|1} \tag{5.30}$$

(e) The row sums of $\mathbf{P}_{n|0}^{-1}$ and $\mathbf{P}_{n|1}^{-1}$ are 1.

Proof. (a): The proof is by induction. For $n = 0$, the identities $\mathbf{P}_{0|1} = \tilde{\mathbf{I}}_0 \mathbf{P}_{0|0} \tilde{\mathbf{I}}_0$ and $\mathbf{P}_{0|0} = \tilde{\mathbf{I}}_0 \mathbf{P}_{0|1} \tilde{\mathbf{I}}_0$ clearly hold. Now suppose that (5.23) and (5.24) are true if n is replaced by $n-1$. Then we have

$$
\tilde{\mathbf{I}}_n \mathbf{P}_{n|0} \tilde{\mathbf{I}}_n = \begin{bmatrix} \mathbf{0} & \tilde{\mathbf{I}}_{n-1} \\ \tilde{\mathbf{I}}_{n-1} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{P}_{n-1|0} & \mathbf{0} \\ \frac{1}{2}\mathbf{P}_{n-1|1} & \frac{1}{2}\mathbf{P}_{n-1|0} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \tilde{\mathbf{I}}_{n-1} \\ \tilde{\mathbf{I}}_{n-1} & \mathbf{0} \end{bmatrix} \tag{5.31}
$$

$$
= \begin{bmatrix} \frac{1}{2}\tilde{\mathbf{I}}_{n-1}\mathbf{P}_{n-1|0}\tilde{\mathbf{I}}_{n-1} & \frac{1}{2}\tilde{\mathbf{I}}_{n-1}\mathbf{P}_{n-1|1}\tilde{\mathbf{I}}_{n-1} \\ \mathbf{0} & \tilde{\mathbf{I}}_{n-1}\mathbf{P}_{n-1|0}\tilde{\mathbf{I}}_{n-1} \end{bmatrix}
$$

$$
= \begin{bmatrix} \frac{1}{2}\mathbf{P}_{n-1|1} & \frac{1}{2}\mathbf{P}_{n-1|0} \\ \mathbf{0} & \mathbf{P}_{n-1|1} \end{bmatrix} \tag{5.32}
$$

$$
= \mathbf{P}_{n-1|1}, \tag{5.33}
$$

where (5.31) and (5.33) are due to the recursive expressions (5.1) and (5.2) while (5.32) follows from the induction hypothesis. It remains to show (5.24). But (5.24) is a direct consequence of the just proven equation and using the identity $\tilde{\mathbf{I}}_n \tilde{\mathbf{I}}_n = \mathbf{I}_n$.

(b): Follows immediately from (a) and the identity $\tilde{\mathbf{I}}_n \tilde{\mathbf{I}}_n = \mathbf{I}_n$.

(c): Starting with the definition of $\mathbf{h}_{n|1}$, we have

$$
\mathbf{h}_{n|1} = -\left(\mathbf{P}_{n|1} \circ \log_2 \mathbf{P}_{n|1} \right) \mathbf{1}_n
$$

$$
= -\left[\left(\tilde{\mathbf{I}}_n \mathbf{P}_{n|0} \tilde{\mathbf{I}}_n \right) \circ \log_2 \left(\tilde{\mathbf{I}}_n \mathbf{P}_{n|0} \tilde{\mathbf{I}}_n \right) \right] \mathbf{1}_n \tag{5.34}
$$

$$
= -\tilde{\mathbf{I}}_n \left(\mathbf{P}_{n|0} \circ \log_2 \mathbf{P}_{n|0} \right) \tilde{\mathbf{I}}_n \mathbf{1}_n \tag{5.35}
$$

$$
= \tilde{\mathbf{I}}_n \mathbf{h}_{n|0},
$$

where (5.34) and (5.35) hold because of (5.23) and Lemma 5.2, respectively.

Equation (5.28) follows from (5.27) and the identity $\tilde{\mathbf{I}}_n \tilde{\mathbf{I}}_n = \mathbf{I}_n$.

(d): Starting with the definition of $\boldsymbol{\omega}_{n|1}$, we have

$$
\begin{aligned}
\boldsymbol{\omega}_{n|1} &= -\mathbf{P}_{n|1}^{-1}\mathbf{h}_{n|1} \\
&= -\tilde{\mathbf{I}}_n \mathbf{P}_{n|0}^{-1}\mathbf{h}_{n|0} \\
&= \tilde{\mathbf{I}}_n \boldsymbol{\omega}_{n|0},
\end{aligned}
\tag{5.36}
$$

where (5.36) follows by replacing $\mathbf{P}_{n|1}$ and $\mathbf{h}_{n|1}$ with (5.23) and (5.27), respectively, and using the identity $\tilde{\mathbf{I}}_n \tilde{\mathbf{I}}_n = \mathbf{I}_n$.

Equation (5.30) follows from (5.29) and the identity $\tilde{\mathbf{I}}_n \tilde{\mathbf{I}}_n = \mathbf{I}_n$.

(e): A standard way to compute $\mathbf{P}_{n|0}^{-1}$ is by Gauss-Jordan elimination. That is, a sequence of elementary row operations is applied to the augmented matrix $\begin{bmatrix} \mathbf{P}_{n|0} & \mathbf{I}_n \end{bmatrix}$ such that $\begin{bmatrix} \mathbf{I}_n & \mathbf{P}_{n|0}^{-1} \end{bmatrix}$ eventually results. Clearly, $\mathbf{P}_{n|0}$ and $\mathbf{I}_n$ are stochastic matrices, i.e., all row sums are equal to one. Thus, at each stage of performing elementary row operations, the row sum of the left matrix equals the row sum of the right matrix. In particular, $\mathbf{P}_{n|0}^{-1}$ has the same row sum as $\mathbf{I}_n$. The same arguments hold for $\mathbf{P}_{n|1}^{-1}$. ∎

We can now state the recursive law for the conditional entropy vector.

Lemma 5.4. For $n \geq 1$, $\mathbf{h}_{2n+2|0}$ satisfies the recursion

$$
\mathbf{h}_{2n+2|0} = \begin{bmatrix}
\mathbf{h}_{2n|0} \\
\frac{1}{2}\mathbf{h}_{2n|0} + \frac{1}{2}\tilde{\mathbf{I}}_{2n}\mathbf{h}_{2n|0} + \mathbf{1}_{2n} \\
\frac{3}{4}\mathbf{h}_{2n|0} + \frac{1}{4}\tilde{\mathbf{I}}_{2n}\mathbf{h}_{2n|0} + \frac{3}{2}\mathbf{1}_{2n} \\
\frac{1}{4}\mathbf{h}_{2n|0} + \frac{3}{4}\tilde{\mathbf{I}}_{2n}\mathbf{h}_{2n|0} + \frac{3}{2}\mathbf{1}_{2n}
\end{bmatrix}.
\tag{5.37}
$$

The initial value for $n = 0$ is given by $\mathbf{h}_{0|0} = 0$.

Before proving Lemma 5.4, we remark that in order to refer to the ith subvector, $1 \leq i \leq 4$, of the conditional entropy vector $\mathbf{h}_{2n+2|0}$, i.e., the subvector composed of the $((i-1) \cdot 2^{2n} + 1)^{\text{th}}$ to the $(i \cdot 2^{2n})^{\text{th}}$ element, we use the superscript (i). For instance, $\mathbf{h}_{2n+2|0}^{(2)}$ refers to $\frac{1}{2}\mathbf{h}_{2n|0} + \frac{1}{2}\tilde{\mathbf{I}}_{2n}\mathbf{h}_{2n|0} + \mathbf{1}_{2n}$. The same notation is used for the weighted conditional entropy vector $\boldsymbol{\omega}_{2n+2|0}$.

Proof of Lemma 5.4. The initial value $\mathbf{h}_{0|0}$ can be computed using $\mathbf{P}_{0|0} = 1$ in (5.14). To show (5.37), we replace $\mathbf{P}_{2n+2|0}$ in (5.14) with (5.17) and compute each of the four entries of the resulting vector. Clearly, $\mathbf{h}_{2n+2|0}^{(1)} = -\left(\mathbf{P}_{2n|0} \circ \log_2 \mathbf{P}_{2n|0}\right)\mathbf{1}_{2n} = \mathbf{h}_{2n|0}$. The three remaining terms are

$$
\begin{aligned}
\mathbf{h}_{2n+2|0}^{(2)} &= \left[-\frac{1}{2}\mathbf{P}_{2n|1} \circ \log_2 \left(\frac{1}{2}\mathbf{P}_{2n|1}\right) - \frac{1}{2}\mathbf{P}_{2n|0} \circ \log_2 \left(\frac{1}{2}\mathbf{P}_{2n|0}\right)\right]\mathbf{1}_{2n} \\
&= \left[\frac{1}{2}\mathbf{P}_{2n|1} - \frac{1}{2}\left(\tilde{\mathbf{I}}_{2n}\mathbf{P}_{2n|0}\tilde{\mathbf{I}}_{2n}\right) \circ \log_2 \left(\tilde{\mathbf{I}}_{2n}\mathbf{P}_{2n|0}\tilde{\mathbf{I}}_{2n}\right) + \frac{1}{2}\mathbf{P}_{2n|0}\right. \\
&\qquad \left. -\frac{1}{2}\mathbf{P}_{2n|0} \circ \log_2 \mathbf{P}_{2n|0}\right]\mathbf{1}_{2n} \\
&= \mathbf{1}_{2n} - \frac{1}{2}\tilde{\mathbf{I}}_{2n}\left(\mathbf{P}_{2n|0} \circ \log_2 \mathbf{P}_{2n|0}\right)\mathbf{1}_{2n} + \frac{1}{2}\mathbf{h}_{2n|0} \\
&= \frac{1}{2}\mathbf{h}_{2n|0} + \frac{1}{2}\tilde{\mathbf{I}}_{2n}\mathbf{h}_{2n|0} + \mathbf{1}_{2n};
\end{aligned}
$$

$$\text{(5.38)}$$
$$\text{(5.39)}$$

$$
\begin{aligned}
\mathbf{h}_{2n+2|0}^{(3)} &= \left[-\frac{1}{4}\mathbf{P}_{2n|1} \circ \log_2 \left(\frac{1}{4}\mathbf{P}_{2n|1}\right) - \frac{1}{4}\mathbf{P}_{2n|0} \circ \log_2 \left(\frac{1}{4}\mathbf{P}_{2n|0}\right)\right. \\
&\qquad \left. -\frac{1}{2}\mathbf{P}_{2n|0} \circ \log_2 \left(\frac{1}{2}\mathbf{P}_{2n|0}\right)\right]\mathbf{1}_{2n} \\
&= \left[\frac{1}{2}\mathbf{P}_{2n|1} - \frac{1}{4}\left(\tilde{\mathbf{I}}_{2n}\mathbf{P}_{2n|0}\tilde{\mathbf{I}}_{2n}\right) \circ \log_2 \left(\tilde{\mathbf{I}}_{2n}\mathbf{P}_{2n|0}\tilde{\mathbf{I}}_{2n}\right) + \mathbf{P}_{2n|0}\right. \\
&\qquad \left. -\frac{3}{4}\mathbf{P}_{2n|0} \circ \log_2 \mathbf{P}_{2n|0}\right]\mathbf{1}_{2n} \\
&= \frac{3}{2}\mathbf{1}_{2n} - \frac{1}{4}\tilde{\mathbf{I}}_{2n}\left(\mathbf{P}_{2n|0} \circ \log_2 \mathbf{P}_{2n|0}\right)\mathbf{1}_{2n} + \frac{3}{4}\mathbf{h}_{2n|0} \\
&= \frac{3}{4}\mathbf{h}_{2n|0} + \frac{1}{4}\tilde{\mathbf{I}}_{2n}\mathbf{h}_{2n|0} + \frac{3}{2}\mathbf{1}_{2n};
\end{aligned}
$$

$$\text{(5.40)}$$
$$\text{(5.41)}$$

$$
\begin{aligned}
\mathbf{h}_{2n+2|0}^{(4)} &= \left[-\frac{1}{2}\mathbf{P}_{2n|1} \circ \log_2 \left(\frac{1}{2}\mathbf{P}_{2n|1}\right) - \frac{1}{4}\mathbf{P}_{2n|1} \circ \log_2 \left(\frac{1}{4}\mathbf{P}_{2n|1}\right)\right. \\
&\qquad \left. -\frac{1}{4}\mathbf{P}_{2n|0} \circ \log_2 \left(\frac{1}{4}\mathbf{P}_{2n|0}\right)\right]\mathbf{1}_{2n} \\
&= \left[\mathbf{P}_{2n|1} - \frac{3}{4}\left(\tilde{\mathbf{I}}_{2n}\mathbf{P}_{2n|0}\tilde{\mathbf{I}}_{2n}\right) \circ \log_2 \left(\tilde{\mathbf{I}}_{2n}\mathbf{P}_{2n|0}\tilde{\mathbf{I}}_{2n}\right)\right. \\
&\qquad \left. +\frac{1}{2}\mathbf{P}_{2n|0} - \frac{1}{4}\mathbf{P}_{2n|0} \circ \log_2 \mathbf{P}_{2n|0}\right]\mathbf{1}_{2n} \\
&= \frac{3}{2}\mathbf{1}_{2n} - \frac{3}{4}\tilde{\mathbf{I}}_{2n}\left(\mathbf{P}_{2n|0} \circ \log_2 \mathbf{P}_{2n|0}\right)\mathbf{1}_{2n} + \frac{1}{4}\mathbf{h}_{2n|0}
\end{aligned}
$$

$$\text{(5.42)}$$
$$\text{(5.43)}$$

$$= \frac{1}{4}\mathbf{h}_{2n|0} + \frac{3}{4}\tilde{\mathbf{I}}_{2n}\mathbf{h}_{2n|0} + \frac{3}{2}\mathbf{1}_{2n}.$$

Observe that (5.38), (5.40), (5.42) follow from using (5.23) and

$$\log_2\left(\frac{1}{2^r}\mathbf{P}_{2n|s_0}\right) = \log_2\left(\frac{1}{2^r}\mathbf{1}_{2n\times 2n} \circ \mathbf{P}_{2n|s_0}\right) = -r\mathbf{1}_{2n\times 2n} + \log_2\mathbf{P}_{2n|s_0}, \quad r = 1, 2.$$

Summing up the scaled vectors $\mathbf{P}_{2n|0}\mathbf{1}_{2n}$ and $\mathbf{P}_{2n|1}\mathbf{1}_{2n}$ in (5.38), (5.40), (5.42) yields the first term in (5.39), (5.41), (5.43). Finally, the second term in (5.39), (5.41),(5.43) follows because it does not matter whether the Hadamard product and the elementwise logarithm is applied before or after permuting the elements of $\mathbf{P}_{2n|0}$ (see Lemma 5.2). ∎

Lemma 5.5. (a) For $n \in \mathbb{N}_0$, $\boldsymbol{\omega}_{2n|0}$ satisfies the recursion

$$\boldsymbol{\omega}_{2n+2|0} = \begin{bmatrix} \boldsymbol{\omega}_{2n|0} \\ \boldsymbol{\omega}_{2n|0} - 2\cdot\mathbf{1}_{2n} \\ \boldsymbol{\omega}_{2n|0} - 2\cdot\mathbf{1}_{2n} \\ \boldsymbol{\omega}_{2n|0} \end{bmatrix} \tag{5.44}$$

with initial value $\boldsymbol{\omega}_{0|0} = 0$.

(b) For $n \in \mathbb{N}$, $\boldsymbol{\omega}_{2n+1|0}$ satisfies the recursion

$$\boldsymbol{\omega}_{2n+1|0} = \begin{bmatrix} \boldsymbol{\omega}_{2n-1|0} \\ \tilde{\mathbf{I}}_{2n-1}\boldsymbol{\omega}_{2n-1|0} \\ \boldsymbol{\omega}_{2n-1|0} - 2\cdot\mathbf{1}_{2n-1} \\ \tilde{\mathbf{I}}_{2n-1}\boldsymbol{\omega}_{2n-1|0} - 2\cdot\mathbf{1}_{2n-1} \end{bmatrix} \tag{5.45}$$

with initial value $\boldsymbol{\omega}_{1|0} = \begin{bmatrix} 0 & -2 \end{bmatrix}^T$.

Remark 5.1. The weighted conditional entropy vector $\boldsymbol{\omega}_{n|0}$ is a *palindrome* for even $n \in \mathbb{N}_0$, i.e., the vector reads the same backwards as forward.

Proof of Lemma 5.5. (a): We show by induction that (5.44) holds. The case $n = 0$ can be verified using Definition 5.1(b) and noting that $\mathbf{P}_{0|0} = \mathbf{P}_{0|0}^{-1} = 1$. Now assume

that (5.44) holds for $n-1$. In order to show (5.44) for n, we evaluate $\boldsymbol{\omega}_{2n+2|0}$ using (5.15) and replacing $\mathbf{P}_{2n+2|0}^{-1}$ and $\mathbf{h}_{2n+2|0}$ with (5.19) and (5.37). Then

$$
\boldsymbol{\omega}_{2n+2|0} = \begin{bmatrix} -\mathbf{P}_{2n|0}^{-1}\mathbf{h}_{2n+2|0}^{(1)} \\ \mathbf{P}_{2n|0}^{-1}\left(\mathbf{P}_{2n|1}\mathbf{P}_{2n|0}^{-1}\mathbf{h}_{2n+2|0}^{(1)} - 2\mathbf{h}_{2n+2|0}^{(2)}\right) \\ \mathbf{P}_{2n|0}^{-1}\left(\mathbf{h}_{2n+2|0}^{(2)} - 2\mathbf{h}_{2n+2|0}^{(3)}\right) \\ \mathbf{M}_0\left(-2\mathbf{P}_{2n|1}\mathbf{P}_{2n|0}^{-1}\mathbf{h}_{2n+2|0}^{(1)} + 3\mathbf{h}_{2n+2|0}^{(2)} + 2\mathbf{h}_{2n+2|0}^{(3)}\right) - 4\mathbf{P}_{2n|0}^{-1}\mathbf{h}_{2n+2|0}^{(4)} \end{bmatrix}. \quad (5.46)
$$

Recall from Lemma 5.4 that $\mathbf{h}_{2n+2|0}^{(1)} = \mathbf{h}_{2n|0}$. Hence, by definition, the first entry of (5.46) is equal to $\boldsymbol{\omega}_{2n|0}$. Replacing $\mathbf{h}_{2n+2|0}^{(1)}$ and $\mathbf{h}_{2n+2|0}^{(2)}$ in (5.46) with the corresponding expressions from Lemma 5.4, we obtain

$$
\boldsymbol{\omega}_{2n+2|0}^{(2)} = \mathbf{P}_{2n|0}^{-1}\left(\mathbf{P}_{2n|1}\mathbf{P}_{2n|0}^{-1}\mathbf{h}_{2n|0} - \mathbf{h}_{2n|0} - \tilde{\mathbf{I}}_{2n}\mathbf{h}_{2n|0} - 2\cdot\mathbf{1}_{2n}\right). \quad (5.47)
$$

In order to simplify (5.47), observe that

$$
-\tilde{\mathbf{I}}_{2n}\boldsymbol{\omega}_{2n|0} + \boldsymbol{\omega}_{2n|0} = 0 \quad (5.48)
$$

since $\boldsymbol{\omega}_{2n|0}$ is a palindrome by hypothesis. Further, using (5.15), (5.26) and the relation $\tilde{\mathbf{I}}_{2n}\tilde{\mathbf{I}}_{2n} = \mathbf{I}_{2n}$, (5.48) reads

$$
\mathbf{P}_{2n|0}^{-1}\cdot\mathbf{h}_{2n|0} - \mathbf{P}_{2n|1}^{-1}\tilde{\mathbf{I}}_{2n}\cdot\mathbf{h}_{2n|0} = 0, \quad (5.49)
$$

which becomes, after a right multiplication with $\mathbf{P}_{2n|1}$,

$$
\mathbf{P}_{2n|1}\mathbf{P}_{2n|0}^{-1}\mathbf{h}_{2n|0} - \tilde{\mathbf{I}}_{2n}\cdot\mathbf{h}_{2n|0} = 0. \quad (5.50)
$$

Finally, using (5.50) in (5.47) as well as the definition of $\boldsymbol{\omega}_{2n|0}$ and noting that $2\mathbf{P}_{2n|0}^{-1}\mathbf{1}_{2n} = 2\cdot\mathbf{1}_{2n}$ (since $\mathbf{P}_{2n|0}^{-1}$ is a stochastic matrix by Lemma 5.3 (e)), we obtain

$$
\boldsymbol{\omega}_{2n+2|0}^{(2)} = \boldsymbol{\omega}_{2n|0} - 2\cdot\mathbf{1}_{2n}.
$$

We continue with the third entry of (5.46). After replacing $\mathbf{h}_{2n+2|0}^{(2)}$ and $\mathbf{h}_{2n+2|0}^{(3)}$ in (5.46)

with the corresponding expressions from Lemma 5.4, it immediately follows that $\boldsymbol{\omega}_{2n+2|0}^{(3)} = \boldsymbol{\omega}_{2n|0} - 2 \cdot \mathbf{1}_{2n}$. Regarding the fourth entry in (5.46), we start with the first term in parentheses. Observe that

$$
\begin{aligned}
& -2\mathbf{P}_{2n|1}\mathbf{P}_{2n|0}^{-1}\mathbf{h}_{2n+2|0}^{(1)} + 3\mathbf{h}_{2n+2|0}^{(2)} + 2\mathbf{h}_{2n+2|0}^{(3)} \\
= & -2\left(\mathbf{P}_{2n|1}\mathbf{P}_{2n|0}^{-1}\mathbf{h}_{2n+2|0}^{(1)} - 2\mathbf{h}_{2n+2|0}^{(2)}\right) - \left(\mathbf{h}_{2n+2|0}^{(2)} - 2\mathbf{h}_{2n+2|0}^{(3)}\right) && (5.51) \\
= & -3\mathbf{P}_{2n|0}\left(\boldsymbol{\omega}_{2n|0} - 2 \cdot \mathbf{1}_{2n}\right). && (5.52)
\end{aligned}
$$

Under consideration of the second and third entry of (5.46), the first parentheses of (5.51) equals $-2\mathbf{P}_{2n|0}\boldsymbol{\omega}_{2n+2|0}^{(2)}$ and the second parentheses $\mathbf{P}_{2n|0}\boldsymbol{\omega}_{2n+2|0}^{(3)}$. Hence, equation (5.52) holds since both $\boldsymbol{\omega}_{2n+2|0}^{(2)}$ and $\boldsymbol{\omega}_{2n+2|0}^{(3)}$ are equal to $\boldsymbol{\omega}_{2n|0} - 2 \cdot \mathbf{1}_{2n}$. Using (5.52) in the fourth entry of (5.46), replacing $\mathbf{h}_{2n+2|0}^{(4)}$ with the corresponding expression from Lemma 5.4 and $\mathbf{M}_0$ with its definition $\mathbf{P}_{2n|0}^{-1}\mathbf{P}_{2n|1}\mathbf{P}_{2n|0}^{-1}$, we have

$$
\begin{aligned}
\boldsymbol{\omega}_{2n+2|0}^{(4)} &= \mathbf{P}_{2n|0}^{-1}\left[-3\mathbf{P}_{2n|1}\left(\boldsymbol{\omega}_{2n|0} - 2 \cdot \mathbf{1}_{2n}\right) - \mathbf{h}_{2n|0} - 3\tilde{\mathbf{I}}_{2n}\mathbf{h}_{2n|0} - 6 \cdot \mathbf{1}_{2n}\right] \\
&= 3\mathbf{P}_{2n|0}^{-1}\left(-\mathbf{P}_{2n|1}\boldsymbol{\omega}_{2n|0} - \tilde{\mathbf{I}}_{2n}\mathbf{h}_{2n|0}\right) + 6 \cdot \mathbf{P}_{2n|0}^{-1}\left(\mathbf{P}_{2n|1}\mathbf{1}_{2n} - \mathbf{1}_{2n}\right) \\
& \quad - \mathbf{P}_{2n|0}^{-1}\mathbf{h}_{2n|0} \\
&= -\mathbf{P}_{2n|0}^{-1}\mathbf{h}_{2n|0} \\
&= \boldsymbol{\omega}_{2n|0}.
\end{aligned}
\qquad (5.53)
$$

Observe that the first parentheses of equation (5.53) evaluates to 0 since it is equal to the left hand side of (5.50). Similarly, the second parentheses in (5.53) evaluates to 0 because $\mathbf{P}_{2n|1}$ is a stochastic matrix.

(b): Recall the recursions

$$
\mathbf{P}_{2n+2|0} = \begin{bmatrix} \mathbf{P}_{2n+1|0} & \mathbf{0} \\ \frac{1}{2}\mathbf{P}_{2n+1|1} & \frac{1}{2}\mathbf{P}_{2n+1|0} \end{bmatrix}
\qquad (5.54)
$$

$$
\mathbf{P}_{2n+2|0}^{-1} = \begin{bmatrix} \mathbf{P}_{2n+1|0}^{-1} & \mathbf{0} \\ -\mathbf{P}_{2n+1|0}^{-1}\mathbf{P}_{2n+1|1}\mathbf{P}_{2n+1|0}^{-1} & 2\mathbf{P}_{2n+1|0}^{-1} \end{bmatrix},
\qquad (5.55)
$$

which follow from (5.1) and (5.21). Computing the first 2^{2n+1} entries of $\boldsymbol{\omega}_{2n+2|0}$ (i.e., the

first half), using Definition 5.1(b), (5.54) and (5.55), we obtain

$$
\begin{bmatrix} \boldsymbol{\omega}_{2n+2|0}^{(1)} \\ \boldsymbol{\omega}_{2n+2|0}^{(2)} \end{bmatrix} = \mathbf{P}_{2n+1|0}^{-1} \left(\mathbf{P}_{2n+1|0} \circ \log_2 \mathbf{P}_{2n+1|0} \right) \mathbf{1}_{2n+1}.
\tag{5.56}
$$

By definition, the right hand side of (5.56) is $\boldsymbol{\omega}_{2n+1|0}$. Hence, under consideration of (5.44), we have

$$
\boldsymbol{\omega}_{2n+1|0} = \begin{bmatrix} \boldsymbol{\omega}_{2n|0} \\ \boldsymbol{\omega}_{2n|0} - 2 \cdot \mathbf{1}_{2n} \end{bmatrix}.
\tag{5.57}
$$

It remains to express $\boldsymbol{\omega}_{2n|0}$ in (5.57) in terms of $\boldsymbol{\omega}_{2n-1|0}$. By the same argument as just used, the first half of the vector $\boldsymbol{\omega}_{2n|0}$ equals $\boldsymbol{\omega}_{2n-1|0}$. Since $\boldsymbol{\omega}_{2n|0}$ is a palindrome, the second half of $\boldsymbol{\omega}_{2n|0}$ equals $\tilde{\mathbf{I}}_{2n-1} \cdot \boldsymbol{\omega}_{2n-1|0}$. Hence,

$$
\boldsymbol{\omega}_{2n|0} = \begin{bmatrix} \boldsymbol{\omega}_{2n-1|0} \\ \tilde{\mathbf{I}}_{2n-1} \cdot \boldsymbol{\omega}_{2n-1|0} \end{bmatrix}.
\tag{5.58}
$$

By replacing $\boldsymbol{\omega}_{2n|0}$ in (5.57) with (5.58), we get (5.45). The initial value $\boldsymbol{\omega}_1 = \begin{bmatrix} 0 & -2 \end{bmatrix}^T$ follows from (5.57) and noting that $\boldsymbol{\omega}_{0|0} = 0$. ■

Remark 5.2. The recursions derived in Lemma 5.4 and 5.5 are with respect to initial state $s_0 = 0$. They can be transformed to recursions with respect to initial state $s_0 = 1$ using (5.27) and (5.29).

5.2.4. Proof of the Main Result

In this section, we evaluate (5.11) based on Lemma 5.5. In particular, we find exact solutions to the optimization problem (5.3)-(5.5) for every $n \in \mathbb{N}$.

Theorem 5.1. Consider the convex optimization problem (5.3) to (5.5). The absolute maximum for input blocks of even length $2n$ is

$$
C_{2n}^{\uparrow} = \frac{1}{2} \log_2 \left(\frac{5}{2} \right) \text{ b/u},
\tag{5.59}
$$

where $n \in \mathbb{N}$. For input blocks of odd length $2n - 1$, the absolute maximum is

$$C_{2n-1}^{\uparrow} = \frac{1}{2n - 1} \left[\log_2 \left(\frac{5}{4} \right) + (n - 1) \cdot \log_2 \left(\frac{5}{2} \right) \right] \text{ b/u}, \tag{5.60}$$

where $n \in \mathbb{N}$.

Proof. Without loss of generality, we assume that the initial state is $s_0 = 0$. Recall (5.11), which for input blocks of length $2n - k$ reads

$$C_{2n-k}^{\uparrow} = \frac{1}{2n - k} \log_2 \left[\mathbf{1}_{2n-k}^T \exp_2 \left(\boldsymbol{\omega}_{2n-k|0} \right) \right] \text{ b/u}, \tag{5.61}$$

where $n \in \mathbb{N}$ and $k = 0, 1$. For $n = 1$, a straightforward computation shows using (5.44) and (5.45) in (5.61), that $C_1^{\uparrow} = \log_2 \left(\frac{5}{4} \right)$ b/u and $C_2^{\uparrow} = \frac{1}{2} \log_2 \left(\frac{5}{2} \right)$ b/u. Now assume that (5.59) and (5.60) hold for some n. In particular, suppose that

$$\mathbf{1}_{2n}^T \exp_2 \left(\boldsymbol{\omega}_{2n|0} \right) = \left(\frac{5}{2} \right)^n \tag{5.62}$$

and

$$\mathbf{1}_{2n-1}^T \exp_2 \left(\boldsymbol{\omega}_{2n-1|0} \right) = \frac{5}{4} \left(\frac{5}{2} \right)^{n-1}. \tag{5.63}$$

We now show that (5.59) and (5.60) hold if n is replaced by $n + 1$. Using the recursions derived in Lemma 5.5, we have

$$\begin{aligned} \mathbf{1}_{2n+2}^T \exp_2 \left(\boldsymbol{\omega}_{2n+2|0} \right) &= \mathbf{1}_{2n}^T \left[2 \exp_2 \left(\boldsymbol{\omega}_{2n|0} \right) + 2 \exp_2 \left(\boldsymbol{\omega}_{2n|0} - 2 \cdot \mathbf{1}_{2n} \right) \right] \\ &= \left(2 + 2 \cdot 2^{-2} \right) \mathbf{1}_{2n}^T \exp_2 \left(\boldsymbol{\omega}_{2n|0} \right) \end{aligned} \tag{5.64}$$

and

$$\begin{aligned} \mathbf{1}_{2n+1}^T \exp_2 \left(\boldsymbol{\omega}_{2n+1|0} \right) &= \mathbf{1}_{2n-1}^T \left[\exp_2 \left(\boldsymbol{\omega}_{2n-1|0} \right) + \exp_2 \left(\tilde{\mathbf{I}}_{2n-1} \boldsymbol{\omega}_{2n-1|0} \right) \right. \\ &\quad \left. + \exp_2 \left(\boldsymbol{\omega}_{2n-1|0} - 2 \cdot \mathbf{1}_{2n-1} \right) + \exp_2 \left(\tilde{\mathbf{I}}_{2n-1} \boldsymbol{\omega}_{2n-1|0} - 2 \cdot \mathbf{1}_{2n-1} \right) \right] \\ &= \mathbf{1}_{2n-1}^T \left[2 \exp_2 \left(\boldsymbol{\omega}_{2n-1|0} \right) + 2 \exp_2 \left(\boldsymbol{\omega}_{2n-1|0} - 2 \cdot \mathbf{1}_{2n-1} \right) \right] \tag{5.65} \\ &= \left(2 + 2 \cdot 2^{-2} \right) \mathbf{1}_{2n-1}^T \exp_2 \left(\boldsymbol{\omega}_{2n-1|0} \right). \tag{5.66} \end{aligned}$$

Equation (5.65) holds since $\mathbf{1}_{2n-1}^T \exp_2\left(\tilde{\mathbf{I}}_{2n-1}\boldsymbol{\omega}_{2n-1|0}\right) = \mathbf{1}_{2n-1}^T \exp_2\left(\boldsymbol{\omega}_{2n-1|0}\right)$ due to the fact that a multiplication with $\tilde{\mathbf{I}}_{2n-1}$ just permutes the entries of $\boldsymbol{\omega}_{2n-1|0}$ (see Lemma 5.2). Finally, using (5.64), (5.66) and the induction hypotheses (5.62), (5.63) in (5.61), we obtain

$$
\begin{aligned}
C_{2n+2}^\uparrow &= \frac{1}{2n+2} \log_2\left[\mathbf{1}_{2n+2}^T \exp_2\left(\boldsymbol{\omega}_{2n+2|0}\right)\right] \\
&= \frac{1}{2n+2} \log_2\left[\left(2 + 2 \cdot 2^{-2}\right) \mathbf{1}_{2n}^T \exp_2\left(\boldsymbol{\omega}_{2n|0}\right)\right] \\
&= \frac{1}{2} \log_2\left(\frac{5}{2}\right) \text{ b/u}
\end{aligned}
$$

and

$$
\begin{aligned}
C_{2n+1}^\uparrow &= \frac{1}{2n+1} \log_2\left[\mathbf{1}_{2n+1}^T \exp_2\left(\boldsymbol{\omega}_{2n+1|0}\right)\right] \\
&= \frac{1}{2n+1} \log_2\left[\left(2 + 2 \cdot 2^{-2}\right) \mathbf{1}_{2n-1}^T \exp_2\left(\boldsymbol{\omega}_{2n-1|0}\right)\right] \\
&= \frac{1}{2n+1} \left[\log_2\left(\frac{5}{4}\right) + n \cdot \log_2\left(\frac{5}{2}\right)\right] \text{ b/u.}
\end{aligned}
$$

∎

Remark 5.3. Observe that $\lim_{n\to\infty} C_{2n+1}^\uparrow = \frac{1}{2}\log_2\left(\frac{5}{2}\right)$ b/u where convergence is from below. Hence, we have

$$
\max_{n\in\mathbb{N}} C_n^\uparrow = \frac{1}{2}\log_2\left(\frac{5}{2}\right) \text{ b/u.}
$$

Unfortunately, the distributions corresponding to (5.59) and (5.60) involve negative "probabilities" — otherwise the capacity of the trapdoor channel would have been established. We elaborate this issue in the following remark.

Remark 5.4. Note that the non-negativity of condition (5.9) does not hold for all $k = 1, \ldots, 2^n$. This can be verified as follows. For a trapdoor channel $\mathbf{P}_{n|0}$, condition (5.9) reads in matrix vector notation as

$$
\left[d_k\right]_{1\le k\le 2^n} = \left(\mathbf{P}_{n|0}^{-1}\right)^T \exp_2\left(\boldsymbol{\omega}_n\right). \tag{5.67}
$$

We now compute the second last row of $\left(\mathbf{P}_{n|0}^{-1}\right)^T$ by the following recursive scheme. Applying the matrix inversion lemma in the form of (5.21) to $\mathbf{P}_{n|0}$, which is written as

in (5.1), and subsequently taking the transpose, then replacing the right bottom block of this matrix, which is $2\left(\mathbf{P}_{n-1|0}^{-1}\right)^{T}$, with the just obtained matrix times two (where $n-1$ is replaced by $n-2$), then applying the same procedure to the right bottom block of $2\left(\mathbf{P}_{n-1|0}^{-1}\right)^{T}$ and so on until the right bottom block equals $2^{n-1}\left(\mathbf{P}_{1|0}^{-1}\right)^{T}$ shows that the second last row of $\left(\mathbf{P}_{n|0}^{-1}\right)^{T}$ equals $\begin{bmatrix} 0 & \cdots & 0 & 2^{n-1} & -2^{n-1} \end{bmatrix}$. Further, by Lemma 5.5, the second to last entry and the last entry of $\boldsymbol{\omega}_n$ equals -2 and 0 if $n \in \mathbb{N}$ is even, and -4 and -2 if $n \in \mathbb{N}$ is odd. Inserting into (5.67) yields

$$
d_{2^n-1} = \begin{cases} -3 \cdot 2^{n-3} & \text{if } n \text{ is even} \\ -3 \cdot 2^{n-5} & \text{if } n \text{ is odd.} \end{cases}
$$

Hence, condition (5.9) does not hold for all $k = 1, \ldots, 2^n$.

5.3. Discussion

We focused on the convex optimization problem (5.3) to (5.5) where the feasible set is larger than the probability simplex. An absolute maximum of the n-letter mutual information was established for any $n \in \mathbb{N}$ by using the method of Lagrange multipliers via [52, Theorem 3.3.3]. The same absolute maximum $\frac{1}{2}\log_2\left(\frac{5}{2}\right) \approx 0.6610$ b/u results for all even n and the sequence of absolute maxima corresponding to odd block lengths converges from below to $\frac{1}{2}\log_2\left(\frac{5}{2}\right)$ b/u as the block length increases. Unfortunately, all absolute maxima are attained outside the probability simplex. Hence, instead of establishing the capacity of the trapdoor channel, we have shown only that $\frac{1}{2}\log_2\left(\frac{5}{2}\right)$ b/u is an upper bound on the capacity. To the best of our knowledge, this upper bound is the tightest known bound. Notably, this upper bound is strictly smaller than the feedback capacity [7]. Moreover, the result gives an indirect justification that the capacity of the trapdoor channel is attained on the boundary of the probability simplex.

6

The Trapdoor Channel and Fractal Geometry**

6.1. Introduction

Unlike in previous chapters, we do not focus on the problem of deriving and attaining the capacity of a given problem in this chapter. We rather reinterpret a given information theoretic model, namely the trapdoor channel, in various ways by intentionally not using information theoretic tools. The approach is motivated by the fact that the capacity of the trapdoor channel, an open problem since 1961, seems to be difficult to solve using standard tools from information theory. The considerable effort, e.g. taken in Chapter 5, to solve the optimization problem (5.3) to (5.5) resulted only in an upper bound on the capacity of the trapdoor channel. On the other hand, the trapdoor channel exhibits lots of structure (see Lemma 5.4 and Lemma 5.5), which might give the capacity if exploited properly. In the following, we present two novel views on the trapdoor channel. Based

---
**submitted to *IEEE Trans. Inf. Theory* [51].

on the underlying stochastic matrices (5.1) and (5.2), the trapdoor channel is described geometrically as a fractal or algorithmically as a recursive procedure. By deriving the underlying iterated function system (IFS), we show that the trapdoor channel with input blocks of length n can be regarded as the nth element of a sequence of shapes approximating a fractal. Second, we present an algorithm that fully characterizes the trapdoor channel and resembles the recursion of generating all permutations of a given string.

This chapter is organized as follows. In Section 6.2, we introduce the mathematical background of fractals and, in particular, the notion of an IFS. In Section 6.3, the IFS corresponding to the trapdoor channel is derived. We study the trapdoor channel as a recursive procedure in Section 6.4. Concluding remarks appear in Section 6.5.

6.2. Prerequisites

We review the idea of *iterated function systems* and *fractals*. For a comprehensive introduction to the subject, see e.g. [56]. A fractal is a geometric pattern which exhibits self-similarity at every scale. A systematic way for generating a fractal starts with a complete metric space (M, d). The space to which the fractal belongs is, however, not M but the space of non-empty compact subsets of M, denoted as $\mathcal{H}(M)$. A suitable choice for a metric for $\mathcal{H}(M)$ is the Hausdorff distance $h_d(A, B) \overset{def}{=} \max\{d(A, B), d(B, A)\}$ where $d(A, B) \overset{def}{=} \max_{x \in A} \min_{y \in B} d(x, y)$ and $d(B, A) \overset{def}{=} \max_{x \in B} \min_{y \in A} d(x, y)$, $A, B \in \mathcal{H}(M)$. It is then guaranteed that $(\mathcal{H}(M), h_d)$ is a complete metric space and that every *contraction*[5] $\varphi : M \to M$ on (M, d) becomes a contraction mapping $\varphi : \mathcal{H}(M) \to \mathcal{H}(M)$ on $(\mathcal{H}(M), h_d)$ defined by $\varphi(A) = \{\varphi(x) : x \in A\}$ for all $A \in \mathcal{H}(M)$.

The following definition and theorem provides a method for generating fractals.

Definition 6.1. [56, Chapter 3.7] A *hyperbolic iterated function system (IFS)* consists of a complete metric space (M, d) together with a finite set of contraction mappings $\varphi_n : M \to M$, with contractivity factors s_n for $n = 1, 2, \dots, N$. The notation for the IFS is $\{M; \varphi_n\, n = 1, 2, \dots, N\}$ and its contractivity factor is $s = \max\{s_n : n = 1, 2, \dots, N\}$.

---

[5]Let (M, d) be a metric space. Recall that a mapping $\varphi : M \to M$ is a contraction if there exists a contractivity factor s, $0 < s < 1$, such that $d\left(\varphi(x), \varphi(y)\right) \le s \cdot d(x, y)$ for all $x, y \in M$.

The fixed point of a hyperbolic IFS, also called the *attractor* or *self-similar set* of the IFS, is a (deterministic) fractal and results from iterating the IFS with respect to any $A \in \mathcal{H}(M)$. This is the content of the following theorem.

Theorem 6.1. [56, Chapter 3.7] Let $\{M; \varphi_n\, n = 1, 2, \dots, N\}$ be an IFS with contractivity factor s. Then the transformation $\Phi : \mathcal{H}(M) \rightarrow \mathcal{H}(M)$ defined by

$$\Phi(A) = \bigcup_{n=1}^{N} \varphi_n(A) \tag{6.1}$$

for all $A \in \mathcal{H}(M)$, is a contraction mapping on the complete metric space $(\mathcal{H}(M), h_d)$ with contractivity factor s. Its unique fixed point, $A^\star \in \mathcal{H}(M)$, obeys

$$A^\star = \Phi(A^\star) = \bigcup_{n=1}^{N} \varphi_n(A^\star),$$

and is given by $A^\star = \lim_{k \to \infty} \Phi^{\circ k}(A)$ for any $A \in \mathcal{H}(M)$.

Many well-known fractals, e.g., the *Koch snowflake*, the *Cantor set*, the *Mandelbrot set*, etc., can be generated using Definition 6.1 and Theorem 6.1. A segment of the Mandelbrot set is shown on the cover of the information theory book by Cover and Thomas [30]. Another representative, the *Sierpinski triangle*, is introduced in the following example. We will later see that this fractal is related to the trapdoor channel.

Example 6.1. *(Sierpinski triangle)* Consider the IFS

$$\left\{ [0,1]^2; \varphi_1(x,y) = \left(\frac{x+1}{2}, \frac{y}{2} \right), \varphi_2(x,y) = \left(\frac{x}{2}, \frac{y+1}{2} \right), \varphi_3(x,y) = \left(\frac{x}{2}, \frac{y}{2} \right) \right\}. \tag{6.2}$$

The affine transformations φ_n, $n = 1, 2, 3$, scale any $A \in \mathcal{H}([0,1]^2)$ by a factor of 0.5. Additionally, φ_1 and φ_2 introduce translations by 0.5 into the x- and y-direction, respectively. The Sierpinski triangle is approximated arbitrarily close by iterating $\Phi(A)$ for any $A \in \mathcal{H}([0,1]^2)$. Fig. 6.1 shows the result after performing four iterations of (6.2). The initial shape A in Fig. 6.1a is a triangle with corner points $(0,0), (1,0), (0,1)$ and in Fig. 6.1b a triangle with corner points $(0,0), (1,1), (1,0)$. As one performs more and more iterations, both sets converge to the same attractor $A^\star$.
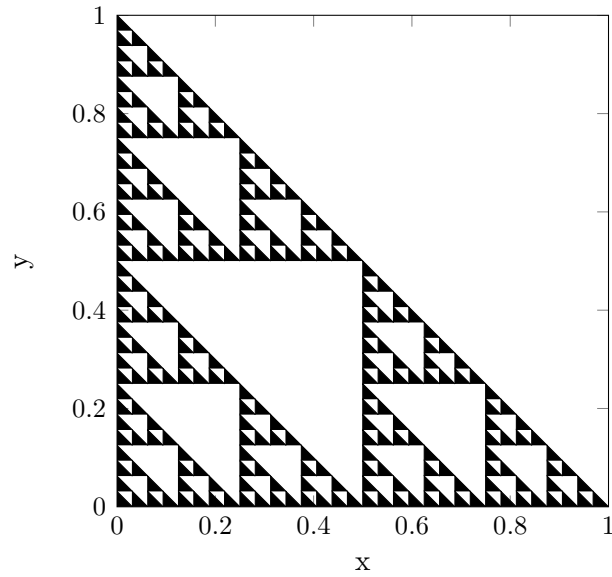
(a) The initial shape is a triangle with corner points $(0,0), (1,0), (0,1)$.
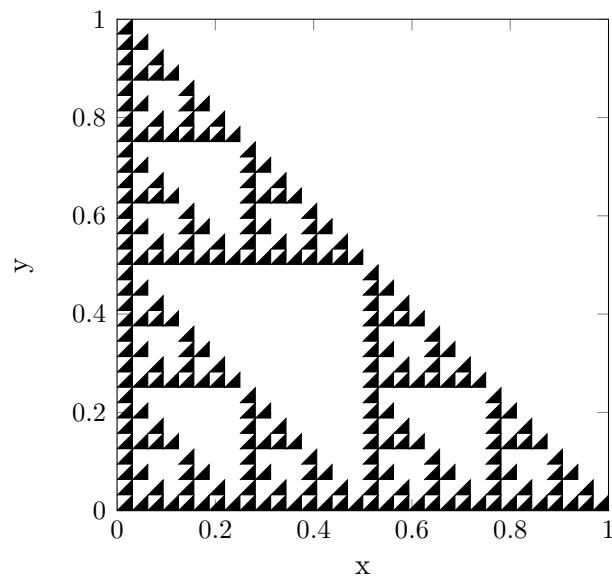
(b) The initial shape is a triangle with corner points $(0,0), (1,1), (1,0)$.

Figure 6.1.: Sierpinski triangle after four iterations.

6.3. The Trapdoor Channel as a Fractal

In this section, we derive a hyperbolic IFS for the trapdoor channel. Instead of working with $\mathbf{P}_{n|s_0}$, we take a geometric approach, i.e., $\mathbf{P}_{n|s_0}$ will be mapped to the unit cube $[0,1]^3 \subset \mathbb{R}^3$.

Definition 6.2. Let $\mathcal{M}$ denote the set $\left\{\mathbf{P}_{n|s_0} : n \in \mathbb{N}_0, s_0 = 0, 1\right\}$ of trapdoor channel matrices. The function $\rho : \mathcal{M} \rightarrow [0,1]^3$ represents each $\mathbf{P}_{n|s_0}$ as a shape in $[0,1]^3$ according to

$$\mathbf{P}_{n|s_0} \mapsto \left(x, y, \left[\mathbf{P}_{n|s_0}\right]_{i,j}\right), \quad \text{for all } 1 \leq i, j \leq 2^n \tag{6.3}$$

where $(i-1) \cdot 2^{-n} < x < i \cdot 2^{-n}$ and $1 - j \cdot 2^{-n} < y < 1 - (j-1) \cdot 2^{-n}$.

Each entry $\left[\mathbf{P}_{n|s_0}\right]_{i,j}$ of $\mathbf{P}_{n|s_0}$ is identified with a square of side length 2^{-n}, which lies at a distance of $\left[\mathbf{P}_{n|s_0}\right]_{i,j}$ from the xy-plane. The alignment of the square corresponding to $\left[\mathbf{P}_{n|s_0}\right]_{i,j}$ with respect to the other squares in $\rho(\mathbf{P}_{n|s_0})$ is in accordance with the alignment of $\left[\mathbf{P}_{n|s_0}\right]_{i,j}$ with respect to the other entries of $\mathbf{P}_{n|s_0}$. Fig. 6.2 depicts the representations $\rho(\mathbf{P}_{1|0})$ and $\rho(\mathbf{P}_{1|1})$. Each of the four regions within $\rho\left(\mathbf{P}_{n|0}\right)$ and $\rho\left(\mathbf{P}_{n|1}\right)$ corresponds to one of the conditional probabilities $0, 0.5$ and 1. The following proposition expresses $\rho\left(\mathbf{P}_{n+1|0}\right)$ and $\rho\left(\mathbf{P}_{n+1|1}\right)$ recursively in terms of $\rho\left(\mathbf{P}_{n|0}\right)$ and $\rho\left(\mathbf{P}_{n|1}\right)$.

Lemma 6.1. The representations $\rho\left(\mathbf{P}_{n+1|0}\right)$ and $\rho\left(\mathbf{P}_{n+1|1}\right)$ of $\mathbf{P}_{n+1|0}$ and $\mathbf{P}_{n+1|1}$ satisfy the recursion laws

$$\rho\left(\mathbf{P}_{n+1|0}\right) = \frac{1}{2} \cdot \left\{\rho\left(\mathbf{P}_{n|0}\right) + \mathbf{e}_x, \ \rho\left(2 \cdot \mathbf{P}_{n|0}\right) + \mathbf{e}_y, \ \rho\left(\mathbf{P}_{n|1}\right)\right\} \tag{6.4}$$

$$\rho\left(\mathbf{P}_{n+1|1}\right) = \frac{1}{2} \cdot \left\{\rho\left(2 \cdot \mathbf{P}_{n|1}\right) + \mathbf{e}_x, \ \rho\left(\mathbf{P}_{n|1}\right) + \mathbf{e}_y, \ \rho\left(\mathbf{P}_{n|0}\right) + \mathbf{e}_x + \mathbf{e}_y\right\} \tag{6.5}$$

for all $n \in \mathbb{N}_0$.

Proof. Recursions (6.4) and (6.5) are a consequence of the structure of (5.1) and (5.2). We just outline the derivation of (6.4). The first term $\frac{1}{2} \cdot \left\{\rho\left(\mathbf{P}_{n|0}\right) + \mathbf{e}_x\right\}$ on the right hand side of (6.4) represents the lower right corner of (5.1). Observe that $\left[\mathbf{P}_{n+1|0}\right]_{i,j}$ is equal to $\frac{1}{2}\left[\mathbf{P}_{n|0}\right]_{i-2^n, j-2^n}$ for all $2^n < i, j, \leq 2^{n+1}$. Hence, scaling the three dimensions of $\rho\left(\mathbf{P}_{n|0}\right)$ by a factor of $\frac{1}{2}$ and shifting the result by $\frac{1}{2}$ along the x-direction yields a
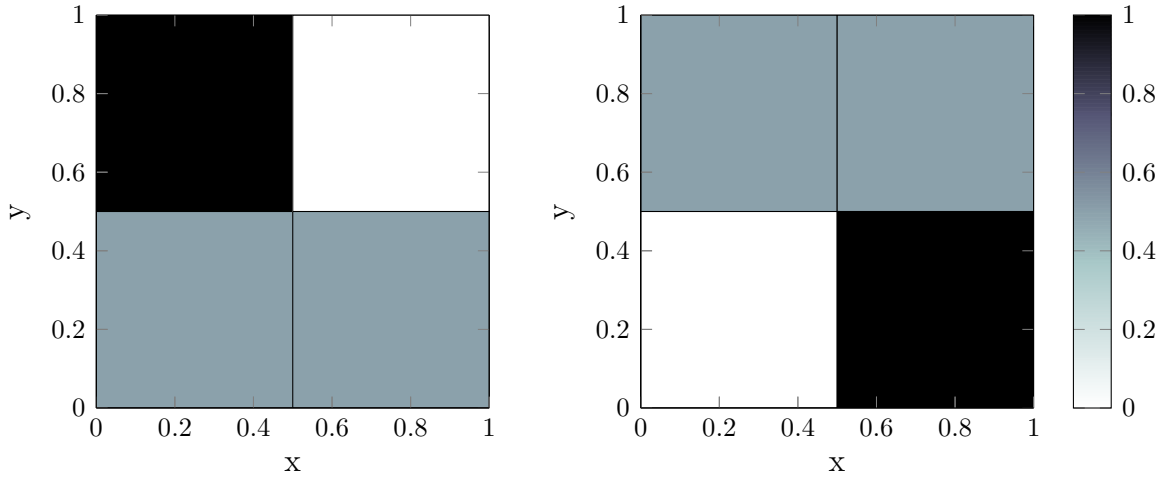
Figure 6.2.: Color map of $\rho(\mathbf{P}_{1|0})$ and $\rho(\mathbf{P}_{1|1})$.

representation of the lower right corner of (5.1) according to Definition 6.2. Similarly, the second term $\frac{1}{2} \cdot \left\{ \rho \left(2 \cdot \mathbf{P}_{n|0} \right) + \mathbf{e}_y \right\}$ of (6.4) represents the upper left corner of (5.1). Note that each entry $\left[\mathbf{P}_{n+1|0} \right]_{i,j}$ is equal to $\left[\mathbf{P}_{n|0} \right]_{i,j}$ for all $1 \leq i, j, \leq 2^n$. Hence, scaling the x- and y-coordinates of $\rho \left(\mathbf{P}_{n|0} \right)$ by a factor of $\frac{1}{2}$ and shifting the resulting figure by $\frac{1}{2}$ along the y-direction yields a representation of the upper left corner $\mathbf{P}_{n|0}$ of (5.1) according to Definition 6.2. The last term $\frac{1}{2} \cdot \rho \left(\mathbf{P}_{n|1} \right)$ of (6.4) represents the lower left corner of (5.1). Clearly, each entry $\left[\mathbf{P}_{n+1|0} \right]_{i,j}$ is equal to $\frac{1}{2} \left[\mathbf{P}_{n|1} \right]_{i-2^n,j}$ for all $2^n < i \leq 2^{n+1}$ and $1 \leq j \leq 2^n$. Hence, scaling all coordinates of $\rho \left(\mathbf{P}_{n|1} \right)$ by a factor of $\frac{1}{2}$ yields a representation of the lower left corner of (5.1) according to Definition 6.2. ∎

Recursions (6.4) and (6.5) are used below to obtain an IFS for the trapdoor channel. Recall from Theorem 6.1 that an IFS is initialized with a single shape. Hence, it would be desirable that the right hand side of (6.4) depends only on $\mathbf{P}_{n|0}$ and the right hand side of (6.5) only on $\mathbf{P}_{n|1}$. The following proposition introduces an affine transformation which turns $\rho \left(\mathbf{P}_{n|0} \right)$ into $\rho \left(\mathbf{P}_{n|1} \right)$ and vice versa.

Lemma 6.2. Let $\tau : [0,1]^3 \to [0,1]^3$ be defined as $\tau(x, y, z) = (-x + 1, -y + 1, z)$. Then

$$\rho \left(\mathbf{P}_{n|1} \right) = \tau \circ \rho \left(\mathbf{P}_{n|0} \right) \tag{6.6}$$

$$\rho\left(\mathbf{P}_{n|0}\right) = \tau \circ \rho\left(\mathbf{P}_{n|1}\right), \tag{6.7}$$

for all $n \in \mathbb{N}_0$.

Proof. Equation (6.7) follows from (6.6) by noting that $\tau \circ \tau = (x, y, z)$. It remains to prove (6.6), which is done by induction. Observe that the affine transformation τ corresponds to a counter-clockwise rotation through 180 degrees about the z-axis and a translation by 1 along the x- and y-direction. Using this property, (6.6) for $n = 1$ is readily verified from Fig. 6.2. Now assume that the assertion holds for some $n > 1$. A direct computation of $\tau \circ \rho\left(\mathbf{P}_{n+1|0}\right)$, using the right hand side of (6.4) and the induction hypotheses (6.6) and (6.7), shows that $\tau \circ \rho\left(\mathbf{P}_{n+1|0}\right)$ is equal to $\rho\left(\mathbf{P}_{n+1|1}\right)$. This is demonstrated for the first function in (6.4). Observe that

$$\begin{aligned}
\tau \circ \frac{1}{2}\left\{\rho\left(\mathbf{P}_{n|0}\right) + \mathbf{e}_x\right\} &= \frac{1}{2}\left\{\left(-x + 1, -y + 1, \left[\mathbf{P}_{n|s_0}\right]_{i,j}\right) + \mathbf{e}_y\right\} \\
&= \frac{1}{2}\left\{\tau \circ \rho\left(\mathbf{P}_{n|0}\right) + \mathbf{e}_y\right\} \\
&= \frac{1}{2}\left\{\rho\left(\mathbf{P}_{n|1}\right) + \mathbf{e}_y\right\},
\end{aligned}$$

where the last step follows from the induction hypothesis. ∎

We can now state the final recursion laws. A combination of Lemma 6.1 and Lemma 6.2, i.e., replacing $\rho\left(\mathbf{P}_{n|1}\right)$ in (6.4) with (6.6), $\rho\left(\mathbf{P}_{n|0}\right)$ in (6.5) with (6.7) and using Definition 6.2, yields the following theorem.

Theorem 6.2. The representations $\rho\left(\mathbf{P}_{n+1|0}\right)$ and $\rho\left(\mathbf{P}_{n+1|1}\right)$ of $\mathbf{P}_{n+1|0}$ and $\mathbf{P}_{n+1|1}$ satisfy the following recursions:

$$\rho\left(\mathbf{P}_{n+1|0}\right) = \begin{cases}
\phi_1(x, y, z) = \left(\dfrac{x+1}{2}, \dfrac{y}{2}, \dfrac{\left[\mathbf{P}_{n|0}\right]_{i,j}}{2}\right), \\[2ex]
\phi_2(x, y, z) = \left(\dfrac{x}{2}, \dfrac{y+1}{2}, \left[\mathbf{P}_{n|0}\right]_{i,j}\right), \\[2ex]
\phi_3(x, y, z) = \left(-\dfrac{x-1}{2}, -\dfrac{y-1}{2}, \dfrac{\left[\mathbf{P}_{n|0}\right]_{i,j}}{2}\right),
\end{cases} \tag{6.8}$$

$$\rho\left(\mathbf{P}_{n+1|1}\right) = \begin{cases} \psi_1(x, y, z) = \left(\dfrac{x+1}{2}, \dfrac{y}{2}, \left[\mathbf{P}_{n|1}\right]_{i,j}\right), \\[3mm] \psi_2(x, y, z) = \left(\dfrac{x}{2}, \dfrac{y+1}{2}, \dfrac{\left[\mathbf{P}_{n|1}\right]_{i,j}}{2}\right), \\[3mm] \psi_3(x, y, z) = \left(-\dfrac{x}{2}+1, -\dfrac{y}{2}+1, \dfrac{\left[\mathbf{P}_{n|1}\right]_{i,j}}{2}\right) \end{cases} \tag{6.9}$$

where $(i-1) \cdot 2^{-n} < x < i \cdot 2^{-n}$ and $1 - j \cdot 2^{-n} < y < 1 - (j-1) \cdot 2^{-n}$ and $1 \le i, j \le 2^n$.

Remark 6.1. The restrictions of ϕ_i and ψ_i, $1 \le i \le 3$, to the x- and y-dimensions are contraction mappings resulting in two hyperbolic IFS with a unique attractor each. An approximation of the attractor for $s_0 = 0$ is shown in the plots on the right side of Fig. 6.3. There is also a relation to the Sierpinski triangle. Observe that ϕ_i and ψ_i, $1 \le i \le 2$, when restricted to the xy-plane, are equal to φ_1, φ_2 in (6.2).

Remark 6.2. Recall that $\mathbf{P}_{0|0} = 1$ and $\mathbf{P}_{0|1} = 1$. Then $\lim_{n \to \infty} \rho\left(\mathbf{P}_{n|s_0}\right)$ for $s_0 = 0$ can be approximated arbitrarily close by iterating (according to Theorem 6.1)

$$\left\{[0,1]^3; \phi_1 = \left(\frac{x+1}{2}, \frac{y}{2}, \frac{z}{2}\right), \phi_2 = \left(\frac{x}{2}, \frac{y+1}{2}, z\right), \phi_3 = \left(-\frac{x-1}{2}, -\frac{y-1}{2}, \frac{z}{2}\right)\right\} \tag{6.10}$$
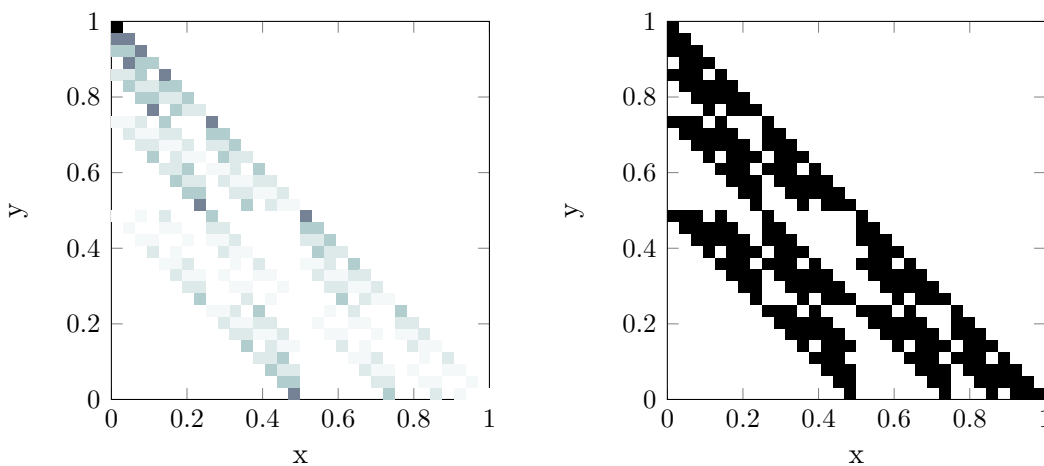
and for $s_0 = 1$

$$\left\{[0,1]^3; \psi_1 = \left(\frac{x+1}{2}, \frac{y}{2}, z\right), \psi_2 = \left(\frac{x}{2}, \frac{y+1}{2}, \frac{z}{2}\right), \psi_3 = \left(-\frac{x}{2}+1, -\frac{y}{2}+1, \frac{z}{2}\right)\right\}, \tag{6.11}$$

where the initial shape can be any $A \in \mathcal{H}([0,1]^3)$ such that the restriction of A to the z-dimension equals one. Fig. 6.3 depicts three, four, and five iterations of (6.10) with an initial shape $\{(x, y, z) \in [0,1]^3 : z = 1\}$.
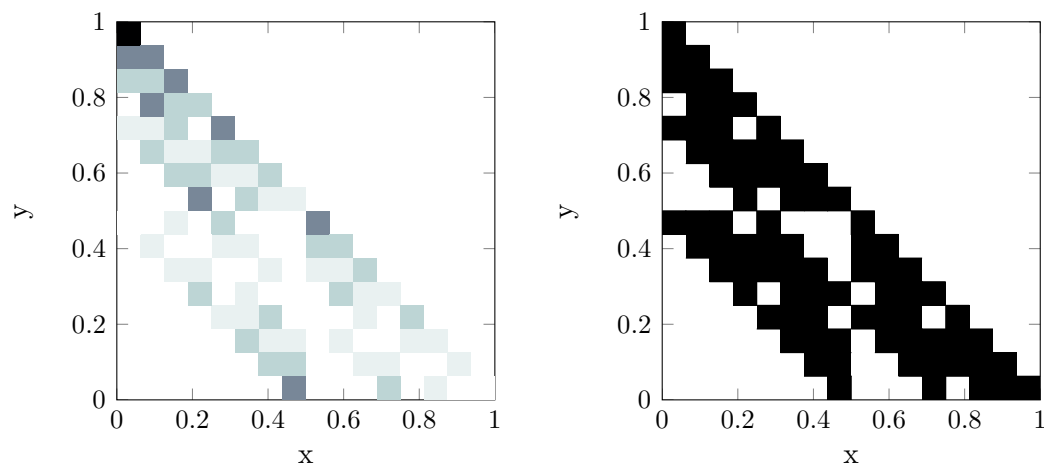
(a) Three iterations.

(b) Four iterations.
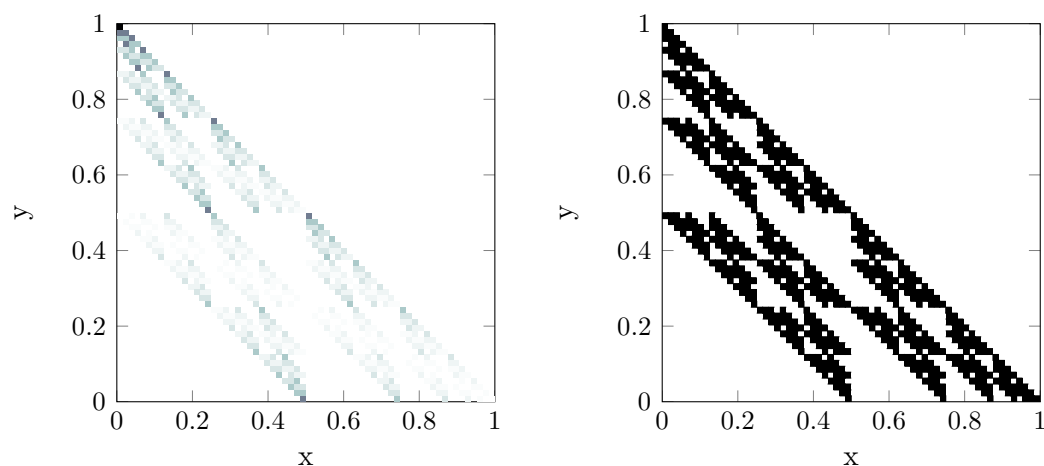
(c) Five iterations.

Figure 6.3.: Three, four, and five iterations of (6.10) and its projections onto the xy-plane. The initial shape is $\{(x, y, z) \in [0, 1]^3 : z = 1\}$. The color scale is the same as in Fig. 6.2.

6.4. Algorithmic View of the Trapdoor Channel

6.4.1. Remarks on the Permutation Nature

The trapdoor channel has been called a permuting channel [54], where the output is a permutation of the input [7]. We point out that in general not all possible permutations of the input are feasible and that not every output is a permutation of the input. The reason that not all permutations are feasible is that the channel actions are causal, i.e., an input symbol at time n cannot become a channel output at a time instance smaller than n. Consider, for instance, the string 101 which, when applied to a trapdoor channel with initial state 0, cannot give rise to an output 110. Next, not every output is a permutation of the input because at a certain time instance the initial state might become an output symbol and, therefore, the resulting output sequence might not be compatible with a permutation of the input. For illustration purposes, consider again the string 101 and initial state 0. Two feasible outputs are 010 and 001, which are not permutations of 110.

6.4.2. The Algorithm

The following recursive procedure GENERATEOUTPUTS computes the set of feasible output sequences and their likelihoods given an input sequence and an initial state. It gives a complete characterization of the trapdoor channel. Generating outputs and their corresponding likelihoods for a particular input sequence might be instrumental for designing codes. In the following, we adopt the standard convention that the first element of a string corresponds to index 0.

> **procedure** GENERATEOUTPUTS($in, out, state, prob$)
>   **if** $in = \emptyset$ **then**
> $set \leftarrow \{out, prob\}$
>   **else if** $in[0] = state$ **then**
> $out \leftarrow out + in[0]$
> $set \leftarrow$ GENERATEOUTPUTS($in.substr(1), out, state, prob$)
>   **else**
> $out \leftarrow out + in[0]$

$\qquad$ $set \leftarrow$ GENERATEOUTPUTS$(in.substr(1), out, state, 0.5 \cdot prob)$

$\qquad$ $out[out.length() - 1] \leftarrow state$ $\qquad$ ▷ $in[0]$ is removed from the end of out

$\qquad$ $set \leftarrow$ GENERATEOUTPUTS$(in.substr(1), out, in[0], 0.5 \cdot prob)$

$\quad$ **end if**

$\quad$ **return** set

end procedure

The four variables $in, out, state, prob$ have the following meaning: in denotes the part of the input string that has not been processed yet, out indicates the part of one particular output string that has been generated so far, $state$ refers to the current channel state, $prob$ denotes the likelihood of out. The procedure is initialized with the complete input string and the initial state of the channel; out is initially empty while $prob$ equals 1. The first if statement checks the simple case of the recursion, namely whether the input string has been processed completely. If yes, the corresponding output out and its likelihood $prob$ is stored and returned in set. Otherwise, we have to distinguish whether the next input symbol $in[0]$ is equal to the current state or not. If yes, the next output takes the value of $in[0]$ with probability 1 (or of $state$, but both are equal), i.e., $out \leftarrow out + in[0]$, and the procedure GENERATEOUTPUTS is applied recursively to the unprocessed part $in.substr(1)$ of the input string, i.e., the substring of in with indices greater than 0. Clearly, $state$ and $prob$ do not change and are passed directly to the recursive call. In the other case, namely when $in[0]$ is not equal to the current state, the next output symbol will have a probability of 0.5 to be either $in[0]$ or $state$. If $in[0]$ becomes the next channel output, the following state remains the same. Then the remaining input string $in.substr(1)$ is processed by the recursive call GENERATEOUTPUTS$(in.substr(1), out, state, 0.5 \cdot prob)$. However, if $state$ becomes the channel output, then the following state will be $in[0]$ and the remaining input string is processed by GENERATEOUTPUTS$(in.substr(1), out, in[0], 0.5 \cdot prob)$.

Observe that a recursive implementation of the algorithm is needed to process inputs of any length, which is not the case if only iterative control structures are used. We emphasize that each of the three recursive calls of the algorithm resembles a recursion for generating all permutations of a string (see, e.g., [57, ch. 8.3]). This gives an algorithmic justification why some output sequences are permutations of the input sequence.

6.5. Discussion

Two different views on the trapdoor channel were presented. We first derived the IFS of the trapdoor channel, which was motivated by the observation that standard approaches from information theory have failed so far to derive its capacity. Subsequently, we described the trapdoor channel by means of a recursive procedure. The procedure, which generates all feasible output sequences and their likelihoods given a particular input sequence, might be helpful to construct codes for the trapdoor channel.

7

Conclusion

7.1. Summary of the Results

In this dissertation, we designed codes for half-duplex-constrained error-free networks and investigated the trapdoor channel. The networks were directed cascades and trees with one or multiple sources. We proposed timing codes which represent information by an information-dependent allocation of the transmission and reception slots of the relays. In addition, we gave several examples demonstrating the coding idea. Based on these codes, we established the (multicast) capacity of cascades and trees with a single source-destination pair and an arbitrary number of intermediate half-duplex-constrained relays. For an unbounded number of relays, we derived an explicit capacity expression. Notably, the capacity in the binary case is logarithm of the golden ratio. In case of cascades with multiple sources, we showed that the strategy achieves the cut-set bound when the rates of the relay sources fall below certain thresholds. We applied this result to a three node line network with two sources for which we found a complete solution.

We further demonstrated that certain well-studied classes of codes, namely erasure

codes and constrained codes, are natural codes in the context of half-duplex-constrained transmission. Specifically, by combining an erasure code with our timing codes, we established the capacity region of half-duplex cascades where the first two nodes are sources. Further, we showed that the multicast capacity of an infinite-depth tree-structured network can be achieved using constrained coding at the source and symbol forwarding at the relays. This approach is interesting from a practical point of view. It is applicable to finite depth trees and has low complexity while being strictly superior to deterministic time-sharing. Examples of corresponding encoders were provided and we showed how to construct them using the state-splitting algorithm.

In the context of the trapdoor channel, we derived an upper capacity bound, namely $\frac{1}{2}\log_2\left(\frac{5}{2}\right)$ b/u. This upper bound shows that feedback increases the capacity of the trapdoor channel. Two recursions resulted along the way, each interesting in its own right. Perhaps these recursions will enjoy further applications on the way of finding the trapdoor channel capacity. Finally, we investigated the fractal nature of the trapdoor channel and derived the underlying iterated function system. Interestingly, the trapdoor channel is related to the Sierpinski triangle.

7.2. Future Directions

We briefly mention some potential directions of research related to this dissertation. In the context of half-duplex networks, one immediate direction would be to consider a network model with noisy transmission links. A promising approach might be to combine the codes proposed in this thesis with error-control techniques. In fact, small alphabet sizes (e.g., binary) should be preferred since the rate gains due to timing are largest for these cases.

As is the case with most engineering solutions, timing codes also exhibit a trade off between performance and complexity. Indeed, by timing sub-blocks of transmission symbols instead of single transmission symbols, the requirements on the synchronization and the switching speed of the relays could be loosened at the expense of timing rate. Therefore, an interesting future direction is to construct and analyze encoders, which use the idea of timing and generate sequences with a minimum number of ON- and OFF-

symbols in a row. In fact, the construction of such encoders should be along the lines of Appendix 4.A.

In Chapter 3, we established the capacity of two special cases of cascades with two sources. We recently noticed that the general case with two sources can be solved using a suitable combination of the timing codes proposed in Chapter 2 and the random codes of Chapter 3. However, finding the capacity region for an arbitrary number of sources seems to be a severe challenge.

The problems studied in the context of half-duplex networks have a strong (lossless) source coding flavor since error-free links are considered. Hence, it might be interesting to interpret our model and results within the framework of source coding and consider reasonable extensions. Further, the channel model and the timing codes proposed in Chapter 2 exhibit similarities with the model and the code proposed in [32] for a memory with defective cells. Hence, there might exist storage applications for which our codes are beneficial.

An interesting question in the context of the trapdoor channel (and of other channels with memory) is whether there is a connection between fractal theory and certain information theoretic properties. It is, e.g., known that graph theory and the zero-error capacity are well-connected [58], [59, Chapter 30].

A
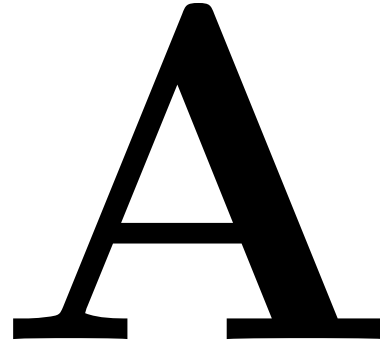
Mathematical Notation and Abbreviations

Notation

N	off symbol indicating no transmission
$\mathcal{Q}$	transmission alphabet
$\mathcal{W}_v$	message set of node v
$|\mathcal{X}|$	cardinality of the set $\mathcal{X}$
$x_{v,k}$	k^{th} entry of the vector $\mathbf{x}_v$
$X_{v,k}$	k^{th} entry of the random vector $\mathbf{X}_v$
$\mathbf{x}_v^n$	the first n entries $(x_{v,1} \ldots, x_{v,n})$ of the vector $\mathbf{x}_v$
$\mathbf{X}_v^n$	the first n entries $(X_{v,1} \ldots, X_{v,n})$ of the random vector $\mathbf{X}_v$
$\mathbf{X}^n$	the random vector $(X_1, \ldots, X_n)$

$w_{v;b}$	a message sent by node v in block b
$w_{0;b-[i;m-1]}$	the messages $\left(w_{0;b-i}, w_{0;b-(i+1)}, \ldots, w_{0;b-(m-1)}\right)$
$\mathbf{x}_v(b)$	a codeword sent by node v in block b
$\mathbf{y}_v(b)$	a codeword received by node v in block b
n_v	number of transmission symbols used by node v per block
k_v	number of transmission symbols used by node v per block to encode own information
S^c	complement of the set S
$\wp(\mathcal{S})$	power set of a set $\mathcal{S}$
$\mathcal{P}_n$	set of all probability n-vectors $\mathbf{p} = (p_1, \ldots, p_n)$
$R_{ts}(q)$	time-sharing rate equal to $0.5 \log_2(q+1)$ b/u
$P_{i;b}^{(n)}$	average probability of a decoding error at node i in block b of length n
ϕ	golden ratio $(1 + \sqrt{5})/2$
$\mathbf{P}_{n\mid s_0}$	$2^n \times 2^n$ matrix characterizing the trapdoor channel with initial state s_0 and input blocks of length n
$\mathbb{N}_0$	natural numbers including 0
$\mathbb{N}$	natural numbers without 0
$\mathbb{R}^n$	the set of ordered n-tuples of real numbers
$\mathbb{R}_{\geq}^n$	the set of ordered n-tuples of non-negative real numbers
$\mathbf{e}_x$	the canonical basis vector $(1, 0, 0)$
$\mathbf{e}_y$	the canonical basis vector $(0, 1, 0)$
$\mathbf{e}_z$	the canonical basis vector $(0, 0, 1)$
$\mathbf{I}_n$	$2^n \times 2^n$ identity matrix
$\tilde{\mathbf{I}}_n$	$2^n \times 2^n$ matrix whose secondary diagonal entries are equal to 1 while the remaining entries are equal to 0
$\mathbf{1}_n$	column vector of length 2^n consisting only of ones
$\mathbf{1}_{n \times n}$	$2^n \times 2^n$ matrix consisting only of ones
$\Phi^{\circ n}$	n-fold composition of the function Φ

$\mathbf{A} \circ \mathbf{B}$ Hadamard product, i.e., the entrywise product, of matrices $\mathbf{A}$ and $\mathbf{B}$

$\log(\cdot)$ logarithm to base e; if applied to a vector or matrix, $\log(\cdot)$ of each element is taken

$\log_2(\cdot)$ logarithm to base 2; if applied to a vector or matrix, $\log_2(\cdot)$ of each element is taken

$\exp_2(\cdot)$ exponential function to base 2; if applied to a vector or matrix, $\exp_2(\cdot)$ of each element is taken

$\left[\mathbf{P}_{n|s_0}\right]_{i,j}$ entry of $\mathbf{P}_{n|s_0}$ in row i and column j

$\mathbf{h}_{n|s_0}$ conditional entropy vector of dimension $2^n \times 1$

$\mathbf{h}_{2n+2|s_0}^{(i)}$ subvector of $\mathbf{h}_{2n+2|s_0}$ from the $((i-1)\cdot 2^{2n}+1)^{\text{th}}$ element to the $(i\cdot 2^{2n})^{\text{th}}$ element, $1 \le i \le 4$

$\boldsymbol{\omega}_{n|s_0}$ weighted conditional entropy vector of dimension $2^n \times 1$

$\boldsymbol{\omega}_{2n+2|s_0}^{(i)}$ subvector of $\boldsymbol{\omega}_{2n+2|s_0}$ from the $((i-1)\cdot 2^{2n}+1)^{\text{th}}$ element to the $(i\cdot 2^{2n})^{\text{th}}$ element, $1 \le i \le 4$

List of Abbreviations

b/sym   bits per symbol
b/u   bits per (channel) use
CF   compress-and-forward
DF   decode-and-forward
FDD   frequency-division duplex
FSC   finite state channel
i.i.d.   independent and identically distributed
IFS   iterated function system
pmf   probability mass function
TDD   time-division duplex

Bibliography

[1] G. Kramer, "Communication strategies and coding for relaying," in *Wireless Communications*, ser. IMA Volumes in Mathematics and its Applications.   New York: Springer, 2007, vol. 143, pp. 163–175.

[2] T. Lutz, "Coding strategies for relay networks with half-duplex constraint," Diplomarbeit, TU M"unchen, 2008.

[3] S. Zander, G. Armitage, and P. Branch, "Covert channels and countermeasures in computer network protocols," *IEEE Commun. Mag.*, vol. 45, pp. 136–142, Dec. 2007.

[4] F. Rieke, D. Warland, R. van Steveninck, and W. Bialek, *Spikes: Exploring the Neural Code.*   MIT Press, 1997.

[5] D. Blackwell, *Information Theory*, E. F. Beckenbach, Ed.   McGraw-Hill Book Co., New York, 1961, vol. Modern Mathematics for the Engineer.

[6] R. G. Gallager, *Information Theory and Reliable Communication.*   John Wiley & Sons, Inc., 1968.

[7] H. H. Permuter, P. Cuff, B. VanRoy, and T. Weissman, "Capacity of the trapdoor channel with feedback," *IEEE Trans. Inf. Theory*, vol. 54, no. 7, pp. 3150–3165, Jul. 2008.

[8] H. H. Permuter, H. Asnani, and T. Weissman, "Capacity of a post channel with and without feedback," 2013, available online: http://arxiv.org/abs/1309.5440.

[9] T. Lutz, C. Hausl, and R. Kötter, "Bits through deterministic relay cascades with half-duplex constraint," *IEEE Trans. Inf. Theory*, vol. 58, pp. 369–381, Jan. 2012.

[10] A. Host Madsen and J. Zhang, "Capacity bounds and power allocation for the wireless relay channel," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 2020–2040, June 2005.

[11] M. A. Khojastepour and A. Sabharwal and B. Aazhang, "On the capacity of 'cheap' relay networks," in *Proc. 37th Annual. Conf. Information Sciences and Systems (CISS), (Baltimore, MD)*, March 12-14, 2003.

[12] S. Toumpis and A. J. Goldsmith, "Capacity regions for wireless ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 736–748, July 2003.

[13] G. Kramer, I. Maric, and R. D. Yates, "Cooperative communications," *Foundations and Trends in Networking*, vol. 1, no. 3-4, pp. 271–425, 2006.

[14] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[15] E. C. van der Meulen, "Three-terminal communication channels," *Adv. Appl. Prob.*, vol. 3, pp. 120–154, 1971.

[16] T. M. Cover and A. A. El Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inf. Theory*, vol. 25, pp. 572–584, Sept. 1979.

[17] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Trans. Inf. Theory*, vol. 51, no. 9, pp. 3037–3063, Sep. 2005.

[18] H. Yamamoto, "Source coding theory for cascade and branching communication system," *IEEE Trans. Inf. Theory*, vol. 27, no. 3, pp. 299–308, May 1981.

[19] Vasudevan, D., Tian, C. and Diggavi, S. N., "Lossy source coding for a cascade communication system with side-informations," in *Proceeding of Allerton Conf. Commun., Control, and Computing, (Monticello, IL)*, Sept. 27 - Sept. 29 2006.

[20] P. Cuff, H. Su, and A. El Gamal, "Cascade multiterminal source coding," in *IEEE Int. Symp. Inf. Theory (ISIT)*, Seoul, Korea, Jun. 28 - Jul.03 2009, pp. 1199–1203.

[21] R. A. Silverman, "On binary channels and their cascade," *IRE Trans. Inf. Theory*, vol. 1, no. 3, pp. 19–27, Dec. 1955.

[22] M. K. Simon, "On the capacity of a cascade of identical discrete memoryless non-singular channels," *IEEE Trans. Inf. Theory*, vol. 16, no. 1, pp. 100–102, Jan. 1970.

[23] U. Niesen, C. Fragouli, and D. Tuninetti, "On the capacity of line networks," *IEEE Trans. Inf. Theory*, vol. 53, no. 11, pp. 4039–4058, Nov. 2007.

[24] A. B. Kiely and J. T. Coffey, "On the capacity of a cascade of channels," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1310–1321, July 1993.

[25] V. Anantharam and S. Verdú, "Bits through queues," *IEEE Trans. Inf. Theory*, vol. 42, pp. 4–18, Jan. 1996.

[26] A. A. Bedekar and M. Azizoğlu, "The information-theoretic capacity of discrete-time queues," *IEEE Trans. Inf. Theory*, vol. 44, pp. 446–461, Mar. 1998.

[27] G. Kramer, "Models and theory for relay channels with receive constraints," in *Proc. 42nd Annual Allerton Conf. on Commun., Control, and Computing*, Monticello, IL, Sep. 29–Oct. 1 2004.

[28] L.-L. Xie and P. R. Kumar, "An achievable rate for the multiple-level relay channel," *IEEE Trans. Inf. Theory*, vol. 51, no. 4, pp. 1348–1358, Apr. 2005.

[29] J. H. van Lint, *Introduction to Coding Theory*. Springer, 1999.

[30] T. M. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[31] P. Vanroose and E. C. van der Meulen, "Uniquely decodable codes for deterministic relay channels," *IEEE Trans. Inf. Theory*, vol. 38, no. 4, pp. 1203–1212, Jul. 1992.

[32] A. V. Kuznetsov and B. S. Tsybakov, "Coding in a memory with defective cells," *Probl. Inf. Transm.*, vol. 10, no. 2, pp. 52–60, 1974.

[33] T. Lutz, G. Kramer, and C. Hausl, "Capacity for half-duplex line networks with two sources," in *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, Jun. 13–18 2010, pp. 2393–2397.

[34] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley, Inc., 1991.

[35] A. E. Gamal and J.-H. Kim, *Network Information Theory.* Cambridge University Press, 2011.

[36] R. Ahlswede and J. Körner, "Source coding with side information and a converse for degraded broadcast channels," *IEEE Trans. Inf. Theory*, vol. 21, no. 6, pp. 629–637, 1975.

[37] F. R. Kschischang and T. Lutz, "A constrained coding approach to error-free half-duplex relay networks," *IEEE Trans. Inf. Theory*, vol. 59, pp. 6258–6260, May 2013.

[38] S. Vijayakumaran, T. F. Wong, and T. M. Lok, "Capacity of the degraded half-duplex relay channel," 2007, available online: http://arxiv.org/abs/0708.2270.

[39] P. Popovski and O. Simeone, "Protocol coding for two-way communications with half-duplex constraints," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Miami, FL, Dec. 6–Dec. 10 2010, pp. 1–5.

[40] L. Zhang, J. Luo, and D. Guo, "Neighbor discovery for wireless networks via compressed sensing," *Perform. Eval.*, vol. 70, no. 7, pp. 457–471, 2013.

[41] D. Guo and L. Zhang, "Virtual full-duplex wireless communication via rapid on-off-division duplex," in *Proc. 48th Annual Allerton Conf. on Commun., Control, and Computing*, Monticello, IL, Sep. 29–Oct. 1 2010.

[42] L. Applebaum, W. U. Bajwa, R. Calderbank, and S. Howard, "Choir codes: Coding for full duplex interference management," in *Proc. 49th Annual Allerton Conf. on Commun., Control, and Computing*, Monticello, IL, Sep. 28–30 2011.

[43] D. E. Lucani and J. Kliewer, "Energy-delay considerations in coded packet flows," in *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, Jul. 31–Aug. 5 2011, pp. 796–800.

[44] L. Zhang and D. Guo, "Capacity of Gaussian channels with duty cycle and power constraints," in *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, Jul. 31–Aug. 5 2011, pp. 513–517.

[45] D. A. Lind and B. H. Marcus, *An Introduction to Symbolic Dynamics and Coding.* Cambridge, 1995.

[46] B. H. Marcus, R. M. Roth, and P. H. Siegel, "An introduction to coding for constrained systems," 2001, available online: http://www.math.ubc.ca/ marcus/Handbook/index.html.

[47] W. G. Bliss, "Circuitry for performing error correction calculations on baseband encoded data to eliminate error propagation," *IBM Tech. Discl. Bull.*, vol. 23, pp. 4633–4634, Mar. 1981.

[48] A. J. van Wijngaarden and K. A. S. Immink, "Maximum runlength-limited codes with error control capabilities," *IEEE J. Select. Areas Commun.*, vol. 19, no. 4, pp. 602–611, Apr. 2001.

[49] H. Y. Chen, M. C. Lin, and Y. L. Ueng, "Low-density parity-check coded recording systems with run-length-limited constraints," *IEEE Trans. Magnetics*, vol. 44, no. 9, pp. 2235–2242, Sep. 2008.

[50] T. Lutz, "Recursions for the trapdoor channel and an upper bound on its capacity," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, Jun. 29–Jul. 4 2014, pp. 2914–2918.

[51] ——, "Various views on the trapdoor channel and an upper bound on its capacity," 2014, available online: http://arxiv.org/abs/1401.4575.

[52] R. Ash, *Information Theory.* Interscience Publishers, 1965.

[53] K. Kobayashi and H. Morita, "An input/output recursion for the trapdoor channel," in *Proc. IEEE Int. Symp. Inf. Theory*, Lausanne, Switzerland, Jun. 30–Jul. 5 2002, p. 423.

[54] R. Ahlswede and A. H. Kaspi, "Optimal coding strategies for certain permuting channels." *IEEE Trans. Inf. Theory*, vol. 33, no. 3, pp. 310–314, 1987.

[55] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed. The John Hopkins University Press, 1996.

[56] M. Barnsley, *Fractals Everywhere.* Academic Press, Inc., 1988.

[57] E. Roberts, *Programming Abstractions in C++.* Prentice Hall, 2014.

[58] L. Lovász, "On the Shannon capacity of a graph," *IEEE Trans. Info. Theory*, vol. 25, pp. 1–7, 1979.

[59] M. Aigner and G. M. Ziegler, *Proofs from THE Book*, 2nd ed. Springer, 2001.