Network Coding:
Theoretical Designs Directed to Applications

Network Coding:
Theoretical Designs Directed to Applications

Mahdi Jafari Siavoshani

EPFL - Ecole Polytechnique Fédérale de Lausanne

Thesis No. 5418 (June 2012)

Thesis presented to the faculty of computer and communication sciences for obtaining the degree of Docteur ès Sciences

Accepted by the jury:

C. Fragouli
Thesis directors

S. N. Diggavi
Expert

R. W. Yeung
Expert

R. Urbanke
Expert

B. Rimoldi
President of the jury

Ecole Polytechnique Fédérale de Lausanne, 2012

To Hamzeh, Azam, and Morteza for all their love, help, and support.

Abstract

The demand for higher throughput and better efficiency are two important challenges for future communication networks. During the past decades, a lot of research studies have been devoted to investigating and proposing near optimal and efficient schemes and algorithms for point-to-point communication. However, in communication networks, especially in wireless systems, we require more intricate algorithms and coding schemes that are optimized for networks rather than for point-to-point communications. In recent years, the *network coding* paradigm has opened new opportunities for network information flow algorithms.

In the first part of the thesis, we consider a non-coherent transmission scenario in a network performing randomized linear network coding. Our main goal is to find the optimal performance in terms of communication rate in such a transmission scenario and to discover the optimal coding scheme to achieve it. It is observed by Koetter et al. [1] that because the network performs an unknown linear transformation, coding over subspaces spanned by the source packets could be a reasonable coding scheme. In order to make this claim information-theoretically justified, we study a multiplicative matrix channel over a finite field with a uniform and independent distribution over the transfer matrix. The capacity for this unicast communication scenario is characterized and it is shown that coding over subspaces is indeed optimal. A similar result is also derived for the two users multiple access problem in such a non-coherent network coding scenario. We then generalize this model by proposing a more universal scenario which is based on an arbitrarily varying channel approach. This result shows the optimality of subspace coding for a wider class of matrix channels, i.e., the channels where only the rank distribution of the transfer matrix is known. Moreover, the above results show that the overhead of schemes based on coding vectors is negligible for many practical situations, i.e., the situations where the rank of the transfer matrix is concentrated around some integer number.

Next, we observe that in a network performing randomized linear network coding, the coding vectors carry topological and state-dependent information about the network. Considering the subspaces spanned by the coding vectors at the relay nodes of the network, we investigate the properties of these subspaces

and leverage them for some practical problems including *network tomography*, *network management*, and *Byzantine attack detection*.

In the last part of the thesis, we consider the problem of secret key sharing among multiple nodes in a network, in the presence of a passive eavesdropper. We assume that there exists a broadcast channel from one of the trusted nodes to the rest of them (including the eavesdropper). Moreover, we assume that the trusted entities can discuss over a public channel overheard by everyone. The secrecy key generation capacity for this problem is still unknown (in general, there exist only some upper and lower bounds). For the erasure broadcast channel as well as the linear deterministic wireless broadcast channel, we propose optimal and efficient schemes that enable arbitrary number of legitimate entities to share a secret key among themselves. By extending these results, we propose achievability schemes for the non-coherent network coding broadcast channel and the state-dependent Gaussian wireless broadcast channel.

Keywords: network coding, randomized linear network coding, matrix channel, non-coherent transmission, subspace coding, channel capacity, arbitrarily varying channel, topology inference, network management, information theoretical secrecy, multi-terminal secret sharing

Riassunto

Migliori velocità di trasmissione ed efficienza sono due importanti sfide per le future reti di comunicazione. Durante gli ultime decenni un importante numero di ricerche è stato dedicato allo studio di codici ed algoritmi per la comunicazione punto-punto. Queste ricerche hanno permesso di trovare schemi efficienti e quasi ottimali, tuttavia nel caso delle reti di comunicazione, in particolare quelle senza fili, questi schemi non sono sufficienti. Negli ultimi anni la tecnica della codifica di rete ha aperto una nuova via per lo sviluppo di algoritmi di flusso dell'informazione di rete più performanti.

La prima parte di questa tesi studia la trasmissione non coerente in reti che effettuano codifica di rete lineare casualizzata. Il principale scopo di questa parte è trovare la velocità di trasmissione ottimale di questo scenario e di trovare un codice che la raggiunga. Koetter et al. in [1] osservano che, visto che in questo scenario la trasformazione lineare effettua dalla rete non è conosciuta, la codifica usando i sottospazi generati dai pacchetti inviati dalla sorgente potrebbe essere un metodo promettente. Allo scopo di giustificare questa osservazione in questa parte della tesi si studia un canale caratterizzato da una matrice moltiplicativa su un campo finito scelta indipendentemente e uniformemente dall'insieme delle matrici possibili: se ne analizza la capacità e si prova che la codifica basata sui sottospazi è effettivamente ottimale. Successivamente un risultato simile è ricavato per il caso dell'accesso multiplo di due utenti alla stessa rete non coerente. Infine viene studiato un modello generalizzato di canale basato sull'approccio dei canali con cambiamenti arbitrari. Questo risultato mostra l'ottimalità della codifica basata sui sottospazi per una più ampia classe di matrici di canale, ovvero dei canali di cui si conosce solo la distribuzione del rango della matrice di trasferimento. Il risultato inoltre mostra che l'informazione di servizio dei codici basati su vettori di codifica è in pratica irrilevante se il rango della matrice di trasferimento è concentrata attorno ad un numero intero.

La seconda parte della tesi si basa sull'osservazione che in un a rete che effettua codifica lineare casualizzata i vettori di codifica contengono informazione sullo stato e sulla topologia della rete. In questa parte si investigano le proprietà dei sottospazi generati dai vettori di codifica ricevuti dai nodi interni alla rete e le si sfruttano per proporre soluzioni ad alcuni problemi pratici come la

tomografia di rete, la gestione di rete e l'individuazione di attacchi bizantini.

L'ultima parte della tesi studia il problema della condivisione di una chiave segreta in un gruppo di nodi in presenza di un aggressore capace di intercettare le comunicazioni. In questa parte si assume che esista un canale broadcast da uno dei nodi fidati verso tutti i nodi della rete, compreso l'aggressore e si assume che i nodi della rete possono comunicare attraverso un canale pubblico. La capacità di generazione di una chiave privata in questo scenario è tuttora sconosciuta (sono conosciute solo maggioranti e minoranti). In questa parte della tesi si presentano degli schemi efficienti ed ottimali che permettono ad un numero arbitrario di nodi di condividere una chiave segreta quando il canale broadcast è un canale con cancellazione. Inoltre, estendendo questi risultati, propone uno schema ottimale per un canale broadcast non coerente e per il canale broadcast senza fili gaussiano.

Parole Chiave: codifica di rete, matrice di canale, comunicazione non coerente, codifica basata su sottospazi, canale con cambiamenti arbitrari, tomografia di rete, gestione di rete, crittografia basata sulla teoria dell'informazione, condivisione di segreti multi-terminale

Acknowledgments [1]

Although most likely acknowledgments are the most fun part of a Ph.D. thesis to read, they are one of the hardest part to write. Now, at the end of my graduate study, I have been recalling all of these years that I spent at EPFL both as a Master's student and a Ph.D. student. I have been remembering all of the enjoyable moments and memorable time I had here that are not possible without so many wonderful friends and colleagues who have been greatly influential in my graduate life. These few paragraphs are an attempt to express my deepest gratitude to all those who made such an exciting experience possible.

First and foremost, my deepest gratitude goes to my adviser, Christina Fragouli, for not only her help and advices as a Ph.D. supervisor, but also for that she was always being to me a source of inspiration and motivation in all aspects of life. It is a great honor for me to be the first Ph.D. student of Christina. This provided me the unique opportunity to spend more time with her, asking questions, discussing the ideas and thinking about the problems together. I have learned a lot from Christina about conduction research, managing a research group, and combining personal and professional life successfully. Moreover, having a very nice, kind, responsible, and supportive character, working under her supervision made my Ph.D. study a great experience full of absolute joy and fun.

I am indebted to Suhas Diggavi for all of his kind and valuable help and support during my study. Although he was not officially my co-adviser, he treated me like one of his students, always available to answer my questions or discuss about various problems.

I am deeply grateful to Bixio Rimoldi, Raymond Yeung, Rudiger Urbanke, and Suhas Diggavi for giving me the honor of having them in my dissertation committee. I also thank them (and Christina) for carefully reading the thesis and their comments on an earlier draft of this work.

I would like to thank Emre Telatar, Rudiger Urbanke, Bixio Rimoldi, Olivier Leveque, Nicolas Macris, and also Christina for providing such a warm and friendly environment we have at EPFL which makes it a place fun to work.

1. This thesis was in part supported by the Swiss National Science Foundation through Grant PP00P2-128639 and Grant PP002-110483, that I gratefully acknowledge.

I am indebted to Francoise Behn for all her help and assistance inside as well as outside EPFL. It is a really great pleasure to have Francoise in the lab, who takes care of all administrative jobs and allows us to focus only on the research. Also, many thanks to Muriel Bardet and Yvonne Huskie for all their help and the great social events of our lab. I would also like to thank Damir Laurenzi and Giovanni Cangiani for their continuous efforts to make our systems running smoothly and to answer our questions with patience.

I would like to extend my warmest gratitude to my dear friend, Amin Karbasi for about sixteen years of friendship. Amin with his special character, with a great social talent, and with an exceptional sense of humor made me feeling at home. During these years our friendship has developed to a great extent and now I should say it has turned to a brotherhood. Additional thanks goes to Soheil Mohajer for all of his kindness and supports. I could always count on his great advises as he was always open to discuss about different subjects varying from research problems to general problems in the life. I would like also to thank Mahdi Cheraghchi for his very unique character and for all the time we spent discussing about different subjects, playing Foosball, or going to restaurant. The same goes to Javad Ebrahimi for his friendship and for all the time we went together for photography or discussed about political issues. A special thank goes to Pedram Pedarsani, who I could always count on his friendship, help, and support. I wish to thank Reza Shokri and Omid Etesami for their deep friendship and for all the great discussions we had which helped me to reorganize my thoughts. Finally, I would like to thank Lorenzo Keller for his very calm and peaceful character which make him a great colleague and a perfect friend. I was always amazed by his vast knowledge about computer which tempted me interrupting him for almost every question that I had; even those silly ones. And he always patiently listen to me and helped me solving my problems.

Here I would like to mention to my former and current colleagues at EPFL who made my PhD time memorable: Abdulkadir Karaagac, Adrian Tarniceriu, Alberto Jimenez, Alla Merzakreeva, Andrei Giurgiu, Ayan Sengupta, Ayfer Ozgur, Christine Neuberg, Christophe Vignat, Cyril Measson, Dinkar Vasudevan, Dominique Tschopp, Eleni Drinea, Emmanuel Abbe, Emre Atsan, Eren Sasoglu, Etienne Perron, Ghid Matouk, I-Hsiang Wang, Iris Safaka, Iryna Andriyanova, Jasper Goseling, Laszlo Czap, Marc Desgroseilliers, Marc Vuffray, Marios Gkatzianas, Marius Kleiner, Mine Aslan, Nick Ruozzi, Nikhil Karamchandani, Peter Berlin, Sanket Dusad, Satish Korada, Shaunak Mishra, Shrinivas Kudekar, Sibi Bhaskaran, Siddhartha Brahma, Stefano Rosati, Uday Pulleti, Vinod Prabhakaran, Vinodh Venkatesan, Vishwambhar Rathi, Vojislav Gajic, and Jeremie Ezri whom we lost two years ago, after a two year battle with cancer.

In addition to those mentioned above, I am grateful to so many amazing friends who made my study in Switzerland an unforgettable stage of my life and full of memorable moments: Ali Ajdari Rad, Ali Hormati, Alireza Roshanghias, Alireza Zobeiri, Amin Jafarian, Amirreza Zobeiri, Arash Golnam, Arash Amini, Armin Tajalli, Azin Amini, Azad Koliji, Ehsan Ardestanizadeh,

Ehsan Kazemi, Elham Ghadiri, Faezeh Malakouti, Fereshteh Bagherimiyab, Ghazale Hosseinabadi, Haleh Chizari, Hamed Alavi, Hamed Hassani, Hamid Khatibi, Hesam Salavati, Hossein Afshari, Hossein Rouhani, Hossein Taghavi, Javad Ebrahimi, Mani Bastani Parizi, Marjan Hamedani, Maryam Davari, Maryam Golbabaee, Maryam Javanmardy, Maryam Zaheri, Masoud Alipour, Milad Maleki, Mina Karzand, Mitra Fatemi, Mohammad Karzand, Mohammad Mahmoody, Mohammd Parhizkar, Mohsen Yousefbeigi, Mona Mahmoudi, Morteza Zadimoghaddam, Narges Radman, Naser Khosropour, Nastaran Asadi Zanjani, Negar Ashari, Nooshin Hadadi, Omid Talebi, Paris Jafari, Parisa Haghani, Pouya Dehghani, Ramtin Pedarsani, Reza Parhizkar, Saeed Haghighat Shoar, Sara Khoshjan, Sara Kherad Pajouh, Sarah Rafiee, Shirin Saeedi, Soonaz Malekzadeh, Tohid Kazerani, Vahid Aref, Vahid Majidzadeh, and Zahra Sinaei. Although, I have spent some time preparing the above list, I am sure I have missed a lot of nice friends on it. I hope that they forgive me!

Above all, I express my heartfelt gratitude to my parents, Hamzeh and Azam, and my brother Morteza who filled my life with joy and happiness. Without their love, support, and patience none of my achievements -in particular this thesis- would have been possible. This thesis is dedicated with love to them.

Contents

"You are at the wheel of your car, waiting at a traffic light, you take the book out of the bag, rip off the transparent wrapping, start reading the first lines. A storm of honking breaks over you; the light is green, you're blocking traffic."

- Italo Calvino

1 Introduction

Communication networks have had a huge impact on the way we are living. With the advent of new technologies and applications, the request for "being connected" keeps increasing dramatically. This implies an increasing demand for higher throughput and more efficient use of communication networks. For example, according to recent network measurements, the IP traffic over the Internet doubles every two years [2, 3]. As another example, Cisco Visual Networking Index Forecast Project [2] predicts a growth of 18-fold for the global mobile Internet data traffic from 2011 to 2016.

Although today we have a good understanding of how to communicate efficiently on point-to-point links, this is not the case for network communications. Most of the today communication techniques are designed to optimize the performance of point-to-point links. As communication networks get more and more complex, we require more elaborate algorithms and coding schemes that are designed and optimized from a network point of view. One of the main objective of the network information theory is to understand how multiple users can share a heterogeneous network efficiently, to characterize the maximum transmission rates, and to explore schemes that can achieve these rates. In this regard, a complete theory of network information would have tremendous impact on design of next generation communication networks. Although we are seemingly far from a complete network information theory, we still need to rethink about our networks' design methodologies.

Network coding (NC) offers a new paradigm for operation of communication networks. This opens new opportunities for network information flow algorithms by changing the perspective we had on how to treat the information bits. In nowadays communication networks, the information bits are treated like commodities. There exist routing algorithms that route the information bits from sources to destinations independently from each other. However, NC

provides this insight that by combining the information bits in the network (i.e., to code them inside the network), in many practical scenarios, we would gain higher transmission rates than when no coding is allowed.

By enabling the relay nodes in a network to code the information, many new questions arise. As a consequence, this area of research has attracted a lot of attention during the past ten years. Some of the main issues in implementing coding in a network are: (i) how to synchronize the node operations in the network, (ii) how the nodes should combine the information they have received, and (iii) how the above tasks can be done using simple operations and in efficient ways.

In large and complex networks, the packets are subjected to random delays, synchronization errors, and they often follow different routes. Hence, it is difficult to implement a centralized algorithm to perform NC. The same difficulties exist for a centralized algorithm to find the network code, i.e., to find the operation of each node in the network. In contrast, *randomized linear NC* [4] has been proposed to provide a simple solution to the above problems. In randomized linear NC, relay nodes randomly and uniformly combine the incoming messages they have received and forward them to their neighbors. In addition to the above-mentioned advantages for randomized linear NC and more importantly, it is shown in [4] that by choosing the nodes operation locally and randomly the multicast capacity of the network can be achieved with high probability.

In this thesis, we mainly focus on communication networks where the relay nodes perform randomized linear NC. We study the unicast and multiple access problem in such a scenario where the source(s) and destination(s) do not know the end-to-end network operation. Although we focus on theoretical problems in this thesis, these are motivated from practical applications.

1.1 Contributions

The main contributions of this thesis are as follows.

- We characterize the capacity of a unicast communication scenario as well as the rate region for a two users multiple access channel (MAC) in a non-coherent NC setup. To this end, we model the overall network operation by a multiplicative matrix channel defined over a finite field and consider two different probabilistic model for the channel transfer matrix. In the first model, we consider that the transfer matrices distributed uniformly at random among all possible matrices and in the second model we study the situation where only the rank distribution of the transfer matrices is known. In both cases we characterize the capacity as well as propose capacity achieving schemes. Our results imply that coding over subspaces, first proposed in [1], is sufficient to achieve the capacity. Moreover, we show that for the most practical situations the rate loss due to the overhead of using coding vectors is negligible.

- We observe that in a randomized linear NC scenario, the message packets traversing the network are not completely arbitrary and carry topological and state-dependent information about the network. In order to extract this information, we study the properties of subspaces spanned by the message packets received at every node and leverage them towards different applications including *network tomography*, *network management*, and *Byzantine attack detection*.
- We study the problem of secret key sharing among multiple trusted (authenticated) nodes in the presence of a passive eavesdropper. We assume that the nodes have access to a broadcast channel overheard by the eavesdropper and are able to discuss over a public channel. In this thesis we focus on different types of broadcast channels. For an *erasure broadcast channel*, we characterize the secrecy capacity and propose an efficient achievability scheme for secret sharing which achieves the capacity. We then extend this result for a *state-dependent deterministic broadcast channel* by characterizing the secrecy capacity and proposing an efficient achievability scheme. By using the above results and applying a nested message set, degraded channel wiretap code, we propose an achievability scheme for a *state-dependent Gaussian broadcast channel*. The proposed scheme achieves the optimal performance for the high dynamic range (where the channel gain differs significantly over different states), high SNR regime. Finally, we consider a *non-coherent NC broadcast channel* and propose an efficient achievability scheme for an arbitrary number of nodes. To this end, we use the insights we gain from studying the problem of secret sharing over an erasure broadcast channel. The proposed scheme is based on subspace coding and we use properties of randomly chosen subspaces developed in Chapter 2.

1.2 Outline

This thesis is organized in three parts. In Part I, we study the optimal transmission rate for a non-coherent unicast NC scenario as well as a non-coherent NC multiple access channel. To this end, in Chapter 3, we model the non-coherent NC channel by a multiplicative matrix channel with uniform distribution over the channel transfer matrices. This model is extended to a wider class of matrix channels in Chapter 4, i.e., channels where only the rank distribution of the transfer matrices is known. As a consequence of Chapters 3 and 4, we find that using coding vectors does not result in a dramatic rate loss if the packet length is not small. However, this result is derived under the assumption that all of the different combinations of the source packets could possibly occur during the information transmission in the network. In Chapter 5, we relax this constraint and consider the situation where each packet traversing the network is a linear combination of a small number of source packets. For this scenario, we show that the overhead of using coding vectors can be reduced by applying an end-to-end coding scheme.

In Part II, using the properties of randomly chosen subspaces stated in Section 2.4, we study the subspaces properties of randomized linear NC. We leverage these properties and adapt them towards different applications including network tomography, network management, and Byzantine attack detection.

In Part III of the thesis, we study the problem of secret key sharing among multiple terminals in the presence of a passive eavesdropper. To this end, in Chapter 7, we focus on wireless environments. First, we model a wireless communication channel by an erasure broadcast channel. Then, we extend this model to a state-dependent deterministic broadcast channel. By using the two previous results, we propose an achievability scheme for a state-dependent Gaussian broadcast channel. In Chapter 8, we study a similar problem but instead we focus on secret sharing among multiple nodes communicating over a network performing randomized linear NC. For this setup, we propose upper and lower bounds for the secrecy capacity. Moreover, the proposed achievability scheme is also efficient.

Finally, we summarize the thesis in Chapter 9. The conclusions are followed by a discussion on various possible directions for future work.

"Science never solves a problem without creating ten more."

- George Bernard Shaw

Background and Some Preliminary Lemmas

2

2.1 Notation

We here introduce the frequently used notation and definitions we are going to use in the following chapters.

Vectors and Matrices over a Finite Field

Let $q \geq 2$ be a power of a prime. Then, we use $\mathbb{F}_q$ to denote the finite field of size q, $\mathbb{F}_q^{m\times n}$ to denote the set of all $m \times n$ matrices over $\mathbb{F}_q$, and $\mathbb{F}_q^L$ to denote the set of all row vectors of length L. The set $\mathbb{F}_q^L$ forms a L-dimensional vector space over the field $\mathbb{F}_q$. Moreover, we use $\mathbb{F}_q^{m\times n,k}$ to denote the set of all $m \times n$ matrices of rank k over $\mathbb{F}_q$.

For a set of vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ we denote their linear span by $\langle \mathbf{v}_1, \ldots, \mathbf{v}_k \rangle$. For a matrix $\boldsymbol{X}$, $\langle \boldsymbol{X} \rangle$ is the subspace spanned by the rows of $\boldsymbol{X}$ and $\langle \boldsymbol{X} \rangle_c$ is the subspace spanned by the columns of $\boldsymbol{X}$. We then have

$$\operatorname{rank}(\boldsymbol{X}) = \dim \langle \boldsymbol{X} \rangle = \dim \langle \boldsymbol{X} \rangle_c . \tag{2.1}$$

Linear Spaces over a Finite Field

Let Π be an arbitrary vector space of finite dimension defined over a finite field $\mathbb{F}_q$. Suppose Π_1 and Π_2 are two subspaces of Π, i.e., $\Pi_1 \sqsubseteq \Pi$ and $\Pi_2 \sqsubseteq \Pi$. We use $\Pi_1 \cap \Pi_2$ to denote the common subspaces of both Π_1 and Π_2 and $\Pi_1 + \Pi_2$ as the smallest subspace that contains both Π_1 and Π_2, namely,

$$\Pi_1 + \Pi_2 = \{\mathbf{v}_1 + \mathbf{v}_2 | \mathbf{v}_1 \in \Pi_1, \mathbf{v}_2 \in \Pi_2\} . \tag{2.2}$$

It is well known that

$$\dim(\Pi_1 + \Pi_2) = \dim(\Pi_1) + \dim(\Pi_2) - \dim(\Pi_1 \cap \Pi_2) . \tag{2.3}$$

We use the following metric to measure the distance between two subspaces,

$$\begin{aligned} d_S(\Pi_1, \Pi_2) &\triangleq \dim(\Pi_1 + \Pi_2) - \dim(\Pi_1 \cap \Pi_2) \\ &= \dim(\Pi_1) + \dim(\Pi_2) - 2\dim(\Pi_1 \cap \Pi_2). \end{aligned} \tag{2.4}$$

In addition to the metric $d_S(\cdot, \cdot)$ defined above, in some cases we will also need a measure that compares how a set $\mathcal{A}$ of subspaces differs from another set $\mathcal{B}$ of subspaces. For this we will use the average pair-wise distance defined as follows

$$D_S(\mathcal{A}, \mathcal{B}) \triangleq \frac{1}{|\mathcal{A}||\mathcal{B}|} \sum_{\pi_a \in \mathcal{A}, \pi_b \in \mathcal{B}} d_S(\pi_a, \pi_b). \tag{2.5}$$

It should be noted that the above relation does not define a metric for the set of subspaces because the self distance of a set with itself is not zero. However, $D_S(\cdot, \cdot)$ satisfies the triangle inequality.

Two subspaces Π_1 and Π_2 are called *orthogonal* if $\Pi_1 \cap \Pi_2 = \{\mathbf{0}\}$. Two subspaces Π_1 and Π_2 of Π are called *complementary* if they are orthogonal and $\Pi_1 + \Pi_2 = \Pi$. These definitions can also be extended to more than two subspaces. Multiple subspaces $\Pi_1 \ldots, \Pi_k$ are called orthogonal if

$$\dim(\Pi_1 + \cdots + \Pi_k) = \dim(\Pi_1) + \cdots + \dim(\Pi_k). \tag{2.6}$$

The subspaces $\Pi_1, \ldots, \Pi_k$ of a space Π are called complementary if they are orthogonal and $\Pi_1 + \cdots + \Pi_k = \Pi$.

Now, consider two subspaces Π_1 and Π_2. We define the subtraction of Π_2 from Π_1 by $U = \Pi_1 \backslash_s \Pi_2$ where U is any subspace of Π_1 which is complementary with $\Pi_1 \cap \Pi_2$. Note that, given Π_1 and Π_2, U is not uniquely defined.

Asymptotics

We use the big-O notation which is defined as follows. Let $f(x)$ and $g(x)$ be two functions defined on some subset of the real numbers. We write $f(x) = O\left(g(x)\right)$ as $x \to \infty$, if there exists a positive real number M and a real number x_0 such that $|f(x)| \leq M|g(x)|$ for all $x > x_0$. For the little o notation we use the following definition. We write $f(x) = o(g(x))$ as $x \to \infty$, if for all $\epsilon > 0$ there exists a real number x_0 such that $|f(x)| \leq \epsilon \cdot |g(x)|$ for all $x > x_0$. We use also the big-Ω notation which is defined as follows. We write $f(x) = \Omega\left(g(x)\right)$ as $x \to \infty$, if we have $g(x) = O\left(f(x)\right)$ as $x \to \infty$. Finally, we use the big-Θ notation to denote that a function is bounded both above and below by another function asymptotically. Formally, we write $f(x) = \Theta\left(g(x)\right)$ as $x \to \infty$, if and only if we have $f(x) = O\left(g(x)\right)$ and $f(x) = \Omega\left(g(x)\right)$ as $x \to \infty$.

Graphs

In this thesis, we represent a communication network by a graph $G = (V, E)$ with set of vertices V and set of edges $E \subseteq V \times V$. To every edge we associate a capacity which captures the maximum possible communication rate over that particular edge.

If the graph $G = (V, E)$ is directed, for an arbitrary edge $e = (u, v) \in E$, we denote $\text{head}(e) = v$ and $\text{tail}(e) = u$. For an arbitrary node $v \in V$, we denote $\text{In}(v)$ the set of incoming edges to v and $\text{Out}(v)$ the set of outgoing edges from v.

Definition 2.1. *A cut between two vertices S and R is a set of graph edges whose removal disconnects S from R. A* min-cut *is a cut with the smallest (minimal) value. The* value *of the cut is the sum of the capacities of the edges in the cut.*

For unit capacity edges, the value of a cut equals the number of edges in the cut, and it is sometimes referred to as the *size* of the cut. We will use the term min-cut to refer to both the set of edges and to their total number. Note that there exists a unique min-cut value, and possibly several min-cuts.

Additional Notations

For the convenience of notation, we use $[i : j]$ to denote the set $\{i, i + 1, \ldots, j - 1, j\}$ where $i, j \in \mathbb{Z}$.

We use the symbols "$\succ$" and "$\prec$" to denote the element-wise inequality between vectors and matrices of the same size.

Given random variables $X_1, \ldots, X_m$, we write $X_{1:m}$ to denote $(X_1, \ldots, X_m)$. We use also $X^{t_0:t}$ to denote $(X[t_0], \ldots, X[t])$ where t is the discreet time index. When $t_0 = 1$ we simply write X^t to denote $(X[1], \ldots, X[t])$.

Let $\mathsf{Uni}(\mathcal{M})$ denote the uniform distribution over the set $\mathcal{M}$. For example, we use $\mathsf{Uni}\left(\mathbb{F}_q^L\right)$ to denote the uniform distribution over vectors of length L that are defined over finite field $\mathbb{F}_q$. Also for $m \times n$ matrices over $\mathbb{F}_q$, we use $\mathsf{Uni}\left(\mathbb{F}_q^{m\times n, r}\right)$ to denote the uniform distribution over $m \times n$ matrices with rank r.

2.2 A Brief Introduction to Network Coding

Let $G = (V, E)$ be a graph with set of vertices V and set of edges $E \subseteq V \times V$ representing a communication network. We assume that each edge has unit capacity and to model edges with higher capacity we allow parallel edges.

The main idea behind NC is simple but elegant. The nodes in a communication network combine the information flows instead of just forwarding them. More precisely, consider a relay node $v \in V$ in a network $G = (V, E)$ as depicted in Figure 2.1. In the NC scheme, the transmitted symbols at every output of the node v is a function of the received symbols at v, namely,

$$\mathbf{w}_i^{(v)} = f_i^{(v)}(\mathbf{u}_1, \ldots, \mathbf{u}_{\text{In}(v)}), \qquad i \in [1 : \text{Out}(v)]. \tag{2.7}$$

However, in order to employ NC in practice where real networks can be hugely complex, the operation of functions $f_i^{(v)}$ should be simple. Linear functions are among the simplest operations that can be used to implement NC in a network. More importantly, it is shown in [5] (see also Theorem 2.1) that there is no loss

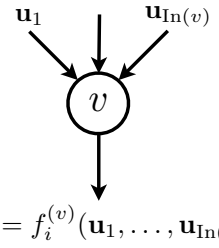


Figure 2.1 – Operation of relay nodes in a network performing NC.

in the transmission rate of a multicast scenario when the nodes operations are linear. In this case we can rewrite (2.7) as follows

$$\mathbf{w}_i^{(v)} = \sum_{j=1}^{\mathrm{In}(v)} \alpha_{i,j}^{(v)} \mathbf{u}_j, \qquad i \in [1 : \mathrm{Out}(v)], \tag{2.8}$$

where $\mathbf{u}_j, \mathbf{w}_i^{(v)} \in \mathbb{F}_q^L$, L is the packet length, $\alpha_{i,j}^{(v)} \in \mathbb{F}_q$, and

$$\boldsymbol{\alpha}_i^{(v)} \triangleq \left[\begin{array}{ccc} \alpha_{i,1}^{(v)} & \cdots & \alpha_{i,\mathrm{In}(v)}^{(v)} \end{array} \right] \in \mathbb{F}_q^{\mathrm{In}(v)} \tag{2.9}$$

is called a *local coding vector*. It is worth to mention that when $L = 1$ the scheme is called *scaler* NC. However, in this thesis we assume that the transmitted and the received symbols are packets over a finite field $\mathbb{F}_q$, namely, we assume $L > 1$.

Suppose that the source node S injects packets $\mathbf{x}_1, \ldots, \mathbf{x}_n$ into the network where $\mathbf{x}_i \in \mathbb{F}_q^L$. Consider the received packet $\mathbf{y}_i^{(R)}$ over the ith incoming edge of the receiver node R. Because the network nodes perform linear operations we can write

$$\mathbf{y}_i^{(R)} = \sum_{j=1}^{n} \beta_{i,j}^{(R)} \mathbf{x}_j, \qquad \forall i \in [1 : \mathrm{In}(R)], \tag{2.10}$$

where $\beta_{i,j}^{(R)} \in \mathbb{F}_q$ and

$$\boldsymbol{\beta}_i^{(R)} \triangleq \left[\begin{array}{ccc} \beta_{i,1}^{(R)} & \cdots & \beta_{i,n}^{(R)} \end{array} \right] \in \mathbb{F}_q^n \tag{2.11}$$

is called a *global coding vector*.

We may also observe that due to the linearity of the network operation, at every node v we have

$$\left\langle \mathbf{w}_1^{(v)}, \ldots, \mathbf{w}_{\mathrm{Out}(v)}^{(v)} \right\rangle \sqsubseteq \left\langle \mathbf{u}_1, \ldots, \mathbf{u}_{\mathrm{In}(v)} \right\rangle, \tag{2.12}$$

and at every receiver R we have

$$\left\langle \mathbf{y}_1^{(R)}, \ldots, \mathbf{y}_{\mathrm{In}(R)}^{(R)} \right\rangle \sqsubseteq \left\langle \mathbf{x}_1, \ldots, \mathbf{x}_n \right\rangle. \tag{2.13}$$

Main Theorems of NC

Now, we briefly review some of the important results of NC. Let us consider a multicast scenario over a network $G = (V, E)$ where c unit rate sources located on the same node S aim to transmit their information to N_r destination nodes $R_1, \ldots, R_{N_r}$. Then, the main theorem of NC [6, 5, 7] can be stated as following.

Theorem 2.1 ([8, Theorem 2.2]). *Consider a directed acyclic graph $G = (V, E)$ with unit capacity edges, c unit rate sources located on the same vertex of the graph and N_r receivers. Assume that the value of the min-cut to each receiver is c. Then there exists a multicast transmission scheme over a large enough finite field $\mathbb{F}_q$, in which intermediate network nodes linearly combine their incoming information symbols over $\mathbb{F}_q$, that delivers the information from the sources simultaneously to each receiver at a rate equal to c.*

If the network topology is known a priori there exist polynomial time algorithms (e.g., see [9, 10, 11, 12]) that *deterministically* construct a linear network code over a sufficiently large field size for a given multicast scenario. However, in most practical applications it is not viable to assume the complete knowledge of the network topology is available. Even if this knowledge is available, it may not be practically possible to apply the deterministic code design schemes due the complexity of the network. In contrast to these deterministic code constructions, a *randomized* scheme is proposed in [4] which is very simple and can be performed in a decentralized manner. This result can be summarized in the following theorem.

Theorem 2.2 ([8, Theorem 5.4]). *Consider an instance multicast scenario from a source node S to receivers in a set $\mathcal{R}$ over a graph $G = (V, E)$ with $N_r = |\mathcal{R}|$ receivers, where the components of local coding vectors are chosen uniformly at random from a field $\mathbb{F}_q$ with $q > N_r$. The probability that all N_r receivers can decode all c sources is at least $(1 - N_r/q)^{\eta'}$, where $\eta' \leq |E|$ is the maximum number of coding points employed by any receiver.*

No matter if the network code is designed a priori using a centralized algorithm or if it is chosen randomly and in a distributed manner, the received packets at a particular node R is a linear transformed version of the transmitted packets by the source S. In particular, if we represent the packets injected by the source S by the rows of a matrix $\boldsymbol{X} \in \mathbb{F}_q^{M \times L}$ and similarly, if we represent the received packets at a node R by the rows of $\boldsymbol{Y}^{(R)} \in \mathbb{F}_q^{N \times L}$, then we can write

$$\boldsymbol{Y}^{(R)} = \boldsymbol{H}^{(R)} \boldsymbol{X}, \tag{2.14}$$

where $\boldsymbol{H}^{(R)} \in \mathbb{F}_q^{N \times M}$ is the channel transfer matrix [1]; it summarizes all of the network operations from the source S to the destination R. If the transfer

1. Whenever it is clear from the context or if it is not very important to explicitly mention to the receiver R, we will remove the superscript "(R)" from the channel transfer matrix $\boldsymbol{H}$ and the received matrix $\boldsymbol{Y}$.

matrix $\boldsymbol{H}^{(R)}$ is fixed and known by the transmitter and receiver, the scheme is called *coherent* NC. Otherwise, it is called *non-coherent* NC. The non-coherent NC is motivated by the decentralized and randomized network code design.

NC in Practice: Coding Vectors vs. Subspace Coding

In practical networks, information is send in packets. Each packet consists of L symbols from a finite field $\mathbb{F}_q$. Coding is performed symbol-wise to each L symbols of every packet.

In large and dynamically changing networks, the packets are subjected to random delays, synchronization errors, and they often follow different routes. It is thus difficult to implement a centralized NC algorithm. To deal with the lack of synchronization, the source packets are grouped into sets called *generations*. Source packets belonging in the same generation are allowed to randomly and in a decentralized manner get combined together, as they traverse the network. Assume that each generation contains M source packets $\{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$.

Implementing randomized NC in practice, is very simple because it does not require any knowledge of the network topology. However, due to its randomized nature, each receiver R does not have access to the channel transfer matrices $\boldsymbol{H}^{(R)}$ (see (2.14)) in order to recover transmitted packets by inverting $\boldsymbol{H}^{(R)}$.

To solve this problem, two different approaches have been proposed so far. The classical *coding vector* approach appends to each source packet $\mathbf{x}_i$ a coding vector $\mathbf{x}_i^C$. Initially, the sources employ as the coding vector

$$\mathbf{x}_i^C = \boldsymbol{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{F}_q^M, \tag{2.15}$$

i.e., $\boldsymbol{e}_i$ has zeros everywhere and 1 is at the ith position. Thus the packets sent by the sources are of the form

$$\mathbf{x}_i = [\boldsymbol{e}_i \mid \mathbf{x}_i^I] \in \mathbb{F}_q^L, \tag{2.16}$$

where $\mathbf{x}_i^I \in \mathbb{F}_q^{L-M}$ is the information part of the transmitted packets. We assumed without loss of generality that the coding vector is placed at the beginning of the packet. Intermediate network nodes perform linear operations on their received packets. In general an arbitrary packet $\boldsymbol{p}$ propagating in the network have the form

$$\boldsymbol{p} = [\boldsymbol{p}^C \mid \boldsymbol{p}^I] \in \mathbb{F}_q^L, \tag{2.17}$$

where $\boldsymbol{p}^I \in \mathbb{F}_q^{L-M}$ is a linear combination of source packets (we call this sometimes information vector), and $\boldsymbol{p}^C \in \mathbb{F}_q^M$ is the coding vector that contains the linear coefficients for the combined source packets.

Each receiver that receives M packets $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M$ with linearly independent coding vectors can recover the original source information. To do so, the receiver

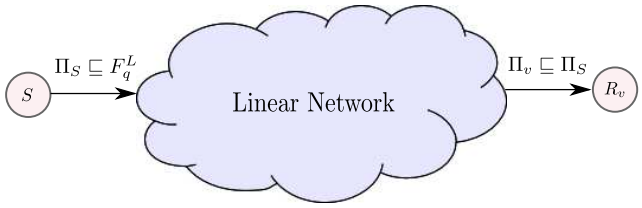


Figure 2.2 – Linear networks preserve the subspaces spanned by the transmitted vectors $\mathbf{x}_1, \ldots, \mathbf{x}_M$, where we have $\Pi_S \triangleq \langle \mathbf{x}_1, \ldots, \mathbf{x}_M \rangle$.

solves the linear equations

$$\begin{bmatrix} \boldsymbol{y}_1^I \\ \boldsymbol{y}_2^I \\ \vdots \\ \boldsymbol{y}_M^I \end{bmatrix} = \underbrace{\begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1M} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{M1} & \beta_{M2} & \cdots & \beta_{MM} \end{bmatrix}}_{\boldsymbol{H} \in \mathbb{F}_q^{M \times M}} \begin{bmatrix} \mathbf{x}_1^I \\ \mathbf{x}_2^I \\ \vdots \\ \mathbf{x}_M^I \end{bmatrix}, \tag{2.18}$$

where the ith row of matrix $\boldsymbol{H}$ is the coding vector corresponding to received packet $\boldsymbol{y}_i$. Since the receiver collects M linearly independent coding vectors, the matrix $\boldsymbol{H}$ is full rank, and thus the original packets can be recovered.

An alternative approach is to use *subspace coding* first proposed in [1]. Subspace coding dispenses of the need to convey coding vectors. In this scheme, neither the receiver(s) nor the source know the channel transfer matrices $\boldsymbol{H}$ in (2.18), i.e., the specific set of linear operations. Sources can only communicate information using subspaces which are unaffected by the linear operations performed on them; see Figure 2.2. Hence, each source uses a *subspace codebook*, i.e., maps each message to a set of vectors that span a different subspace. More precisely, the information transmission is done not via the choice of $\mathbf{x}_i$ but rather by the choice of the vector space spanned by $\{\mathbf{x}_i\}_{i=1}^M$, i.e.,

$$\Pi_S \triangleq \langle \mathbf{x}_1, \ldots, \mathbf{x}_M \rangle . \tag{2.19}$$

In Chapters 3 and 4, we study the non-coherent NC scenario from an information theoretical perspective.

2.3 Grassmanian and Gaussian Coefficient

Definition 2.2 (Grassmannian and Gaussian coefficient [13, 14])**.** *The Grassmannian* $\mathrm{Gr}(L,d)_q$ *is the set of all* d*-dimensional subspaces of the* L*-dimensional space over a finite field* $\mathbb{F}_q$*, namely,*

$$\mathrm{Gr}(L,d)_q \triangleq \big\{ \pi \sqsubseteq \mathbb{F}_q^L : \dim(\pi) = d \big\}. \tag{2.20}$$

The cardinality of $\mathrm{Gr}(L,d)_q$ *is the Gaussian coefficient, namely,*

$$\begin{bmatrix} L \\ d \end{bmatrix}_q \triangleq |\mathrm{Gr}(L,d)_q| = \frac{(q^L-1)\cdots(q^{L-d+1}-1)}{(q^d-1)\cdots(q-1)}. \tag{2.21}$$

Definition 2.3. *We define* $\mathrm{Sp}(L,k)_q$ *to be the set (sphere) of all subspaces of dimension at most* k *in the* L*-dimensional space* $\mathbb{F}_q^L$*, namely,*

$$\mathrm{Sp}(L,k)_q \triangleq \bigcup_{d=0}^{\min[k,L]} \mathrm{Gr}(L,d)_q = \big\{\pi \sqsubseteq \mathbb{F}_q^L : \dim(\pi) \leq \min[k,L]\big\}. \tag{2.22}$$

The cardinality of $\mathrm{Sp}(L,k)_q$ *equals*

$$\mathcal{S}(L,k)_q \triangleq |\mathrm{Sp}(L,k)_q| = \sum_{d=0}^{\min[k,L]} |\mathrm{Gr}(L,d)_q|. \tag{2.23}$$

Definition 2.4. *We denote by* $\psi(L,k,\pi_d)_q$ *the number of different* $k\times L$ *matrices with elements from a field* $\mathbb{F}_q$*, such that their rows span a specific subspace* $\pi_d \sqsubseteq \mathbb{F}_q^L$ *of dimension* $0 \leq d \leq \min[k,L]$*.*

For simplicity, in the rest of the thesis we will drop the subscript q in the previous definitions whenever it is obvious from the context.

Preliminary Lemmas

We here state some preliminary lemmas related to the definitions introduced in Section 2.3.

Existing bounds in the literature allow to approximate the Gaussian number, for example, we have from [1, Lemma 4] that (see also [15, Section III])

$$q^{d(L-d)} < \begin{bmatrix} L \\ d \end{bmatrix} < \frac{q^{d(L-d)}}{\prod_{j=1}^{\infty}(1-q^{-j})} < 4q^{d(L-d)}, \quad \forall d : 0 < d < L. \tag{2.24}$$

Using Definition 2.21 and (2.24) we have Lemma 2.1.

Lemma 2.1. *For large values of* q*, we can approximate the Gaussian number as follows*

$$\begin{bmatrix} L \\ d \end{bmatrix} = q^{d(L-d)}\big(1+O(q^{-1})\big). \tag{2.25}$$

Lemma 2.2. *The following relation for the Gaussian number holds [16, 14]*

$$\begin{bmatrix} L-d \\ D-d \end{bmatrix}\begin{bmatrix} L \\ d \end{bmatrix} = \begin{bmatrix} L \\ D \end{bmatrix}\begin{bmatrix} D \\ d \end{bmatrix}, \tag{2.26}$$

for all $0 \leq d \leq D \leq L$*.*

Lemma 2.3. *The number of different $m \times n$ matrices with rank $0 \leq k \leq \min[m,n]$ over $\mathbb{F}_q$ is equal to [17]*

$$\begin{aligned}\left|\mathbb{F}_q^{m\times n,k}\right| &= q^{(m+n-k)k}\prod_{i=0}^{k-1}\frac{(1-q^{i-n})(1-q^{i-m})}{(1-q^{i-k})}\\ &= \left|\mathbb{F}_q^{k\times m,k}\right|\begin{bmatrix}n\\k\end{bmatrix}\\ &= \left|\mathbb{F}_q^{k\times n,k}\right|\begin{bmatrix}m\\k\end{bmatrix}.\end{aligned} \tag{2.27}$$

By applying [18, Lemma 2] with proper choice of the parameters we have the following lemma.

Lemma 2.4. *Let π_y be a fixed subspace of $\mathbb{F}_q^L$ with dimension d_y. Then the number of different subspaces $\pi_x \sqsubseteq \mathbb{F}_q^L$ with dimension d_x, $d_y \leq d_x \leq L$, that contain π_y is equal to $\begin{bmatrix}L-d_y\\d_x-d_y\end{bmatrix}$.*

Lemma 2.5. *For $\psi(L,k,\pi_d)$ given in Definition 2.4, we have that [16]*

$$\psi(L,k,\pi_d) = \prod_{i=0}^{d-1}(q^k - q^i) = q^{\binom{d}{2}}\prod_{i=0}^{d-1}(q^{k-i}-1), \tag{2.28}$$

i.e., it does not depend on L.

Remark 2.1. *Since $\psi(L,k,\pi_d)$ does not depend on L, and only depends on π_d through its dimension, as a shorthand notation we will also use $\psi(k,d)$ instead of $\psi(L,k,\pi_d)$, where $d = \dim(\pi_d)$.*

Using Lemma 2.5 the following lower and upper bounds are straightforward

$$(1 - dq^{-k+d-1}) < \left(1 - \sum_{i=0}^{d-1} q^{-k+i}\right) < \frac{\psi(k,d)}{q^{kd}} < 1, \tag{2.29}$$

which imply Lemma 2.6 (see also [15]).

Lemma 2.6. *For large values of q the following approximation holds*

$$\psi(k,d) = q^{kd}\big(1 - O(q^{-1})\big). \tag{2.30}$$

It is also worthwhile to mention that $\psi(k,d)\begin{bmatrix}L\\d\end{bmatrix}$ is the number of $k \times L$ matrices of rank d. We can count all the $k \times L$ matrices through the following Lemma 2.7 (also see [13, 14], and [16, Corollary 5]).

Lemma 2.7. *For every $k > 0$ and $L > 0$ we can write*

$$\sum_{d=0}^{\min[k,L]} \psi(k,d)\begin{bmatrix}L\\d\end{bmatrix} = q^{kL}, \tag{2.31}$$

where $\psi(k,0) = 1$.

2.4 Properties of Random Vector Spaces over a Finite Field $\mathbb{F}_q^n$

In this section, we will state and prove some basic properties and results that we will exploit towards various applications in the following chapters. In particular, we will investigate the properties of random sampling from vector spaces over a finite field. Such properties give us a better insight and understanding of randomized NC and form a foundation for the results and algorithms presented in Chapters 6 and 8.

Sampling Subspaces over $\mathbb{F}_q^n$

Here, we explore properties of randomly sampled subspaces from a vector space $\mathbb{F}_q^n$. We start with the following lemma that explores properties of a single subspace.

Lemma 2.8. *Suppose we choose m vectors from an n-dimensional vector space $\Pi_S = \mathbb{F}_q^n$ uniformly at random to construct a subspace Π. Then the subspace Π will be full rank (has dimension $\min[m,n]$) w.h.p. (with high probability)*[2]*, namely,*

$$\mathbb{P}\left[\dim(\Pi) = \min[m,n]\right] = 1 - O\left(q^{-1}\right). \tag{2.32}$$

Proof. For the proof refer to Appendix 2.A. □

We conclude that for large values of q, selecting $m \leq n$ vectors uniformly at random from $\mathbb{F}_q^n$ to construct a subspace Π is equivalent to choosing an m-dimensional subspace from $\mathbb{F}_q^n$ uniformly at random. Note that this is not true for small values of q.

We next examine connections between multiple subspaces.

Lemma 2.9. *Let Π_1 and Π_2 be two subspaces of $\Pi_S = \mathbb{F}_q^n$ with dimension d_1 and d_2 respectively, intersection of dimension d_{12} and $\Pi_1 \nsubseteq \Pi_2$ (i.e., $d_{12} < d_1$). Construct Π_1' by choosing m vectors from Π_1 uniformly at random. Then*

$$\mathbb{P}\left[\Pi_1' \sqsubset \Pi_2\right] = O\left(q^{-m}\right). \tag{2.33}$$

Proof. For the proof refer to Appendix 2.A. □

Lemma 2.10. *Suppose Π_k is a k-dimensional subspace of a vector space $\Pi_S = \mathbb{F}_q^n$. Select m vectors uniformly at random from Π_S to construct the subspace Π. Then w.h.p. we have*

$$\begin{aligned}\dim(\Pi \cap \Pi_k) &= \min\left[k, (m-(n-k))^+\right] \\ &= \left(\min[m,n] + k - n\right)^+. \end{aligned} \tag{2.34}$$

2. Throughout this section, when we talk about an event occurring with high probability, we mean that its probability behaves like $1 - O\left(q^{-1}\right)$, which goes to 1 as $q \to \infty$.

Proof. For the proof refer to Appendix 2.A. □

Corollary 2.1. *Suppose Π_1 and Π_2 are two subspace of $\mathbb{F}_q^n$ with dimension d_1 and d_2 respectively and joint dimension d_{12}. Let us take m_1 vectors uniformly at random from Π_1 and m_2 vectors from Π_2 to construct subspaces $\hat{\Pi}_1$ and $\hat{\Pi}_2$. Then w.h.p. we have*

$$\dim(\hat{\Pi}_1 \cap \hat{\Pi}_2) = \min\Big[d_{12}, (m_1+m_2-(d_1+d_2-d_{12}))^+, \\ (m_1-(d_1-d_{12}))^+, (m_2-(d_2-d_{12}))^+\Big]. \tag{2.35}$$

Proof. For the proof refer to Appendix 2.A. □

By choosing $\Pi_1 = \Pi_2 = \mathbb{F}_q^n$ in Corollary 2.1 we have the following corollary.

Corollary 2.2. *Let us construct two subspaces $\hat{\Pi}_1$ and $\hat{\Pi}_2$ by choosing m_1 and m_2 vectors uniformly at random respectively from $\mathbb{F}_q^n$. Then the subspaces $\hat{\Pi}_1$ and $\hat{\Pi}_2$ will be disjoint w.h.p. if $m_1 + m_2 \leq n$.*

Lemma 2.11. *Suppose that k subspaces $\Pi_1, \ldots, \Pi_k$, with dimensions $d_1, \ldots, d_k$, are chosen independently and uniformly at random from $\mathbb{F}_q^n$. Then with high probability (probability of order $1 - O(q^{-1})$) we have*

$$\dim(\Pi_1 + \cdots + \Pi_k) = \min[d_1 + \cdots + d_k, n], \tag{2.36}$$

and

$$\dim(\Pi_1 \cap \cdots \cap \Pi_k) = [d_1 + \cdots + d_k - (k-1)n]^+. \tag{2.37}$$

Note that if one of the subspaces, for example Π_1, be a fixed subspace then the above results still hold.

Proof. For the proof refer to Appendix 2.A. □

We are now ready to discuss one of the important properties of randomly chosen subspaces which is very useful for our work: randomly selected subspaces tend to be "as far as possible". We will clarify and make precise what we mean by "as far as possible", see also [19]. We first review the definition of a subspace in *general position* with respect to a family of subspaces.

Definition 2.5 ([19, Chapter 3])**.** *Let Π_S be an n-dimensional vector space over the field $\mathbb{F}_q$ and for $i = 1, \ldots, r$, let Π_i be a subspace of Π_S, with $\dim(\Pi_i) = d_i$. A subspace $\Pi \sqsubseteq \Pi_S$ of dimension d is in general position with respect to the family $\{\Pi_i\}_{i=1}^r$ if*

$$\dim(\Pi_i \cap \Pi) = \max[d_i + d - n, 0], \quad \forall i \in \{1, \ldots, r\}. \tag{2.38}$$

It should be noted that $\max[d_i+d-n,0]$ is the minimum possible dimension of $(\Pi_i \cap \Pi)$. So what the above definition says is that the intersection of Π and each Π_i is as small as possible. Using the above definition we can state the following theorem [3].

Theorem 2.3. *Suppose* $\{\Pi_i\}_{i=1}^r$ *are subspaces of* $\Pi_S = \mathbb{F}_q^n$. *Let us construct a subspace* Π *by randomly choosing* m *vectors from* Π_S. *Then* Π *will be in general position with respect to the family* $\{\Pi_i\}$ *w.h.p.*

Proof. For the proof refer to Appendix 2.A. □

Theorem 2.3 demonstrates a nice property of randomized NC where the subspaces spanned by coding vectors (or more generally, the subspace spanned by the transmitted packets) tend to be as far as possible on different paths of the network.

In Chapter 6, we will use these properties of random subspaces towards studying different applications, including topology inference and network management in networks employing NC. To this end, we will derive similar properties but for random subspaces evolving during the time.

3. Different versions of this theorem can be easily derived from results in the literature [19], but we repeat here a short derivation for completeness.

2.A Omitted Proofs

Proof of Lemma 2.8. First, let us fix a basis for Π_S. Then choosing m vector uniformly at random from Π_S is equivalent to choose an $m \times n$ matrix $\boldsymbol{A}$ uniformly at random from $\mathbb{F}_q$ and construct $\Pi = \langle \boldsymbol{A} \rangle$ with respect to this fixed basis.

From Lemma 2.3 we know that the number of different $m \times n$ matrices with rank $0 \leq k \leq \min[m, n]$ over $\mathbb{F}_q$ is equal to

$$\left|\mathbb{F}_q^{m\times n,k}\right| = q^{(m+n-k)k} \prod_{i=0}^{k-1} \frac{(1-q^{i-n})(1-q^{i-m})}{(1-q^{i-k})}. \tag{2.39}$$

So we can write

$$\mathbb{P}\left[\dim(\Pi) = k\right] = \frac{\left|\mathbb{F}_q^{m\times n,k}\right|}{q^{mn}}. \tag{2.40}$$

Then using the Taylor series $\frac{1}{1-\epsilon} = 1+\epsilon+\epsilon^2+\cdots$ for $|\epsilon| < 1$, choosing $\epsilon = q^{-1}$, we can write

$$\Pr[\dim(\Pi) = k] = q^{-(m-k)(n-k)}[1 - O(q^{-1})]. \tag{2.41}$$

By setting $k = \min[m, n]$ we are done. □

Proof of Lemma 2.9. The probability that all m vectors are in the intersection is

$$\mathbb{P}\left[\Pi_1' \sqsubset \Pi_2\right] = \left(\frac{q^{d_{12}}}{q^{d_1}}\right)^m = q^{(d_{12}-d_1)m}, \tag{2.42}$$

which is of order $O\left(q^{-m}\right)$ provided that $\Pi_1 \nsubseteq \Pi_2$, i.e., $d_{12} < d_1$. □

Proof of Lemma 2.10. Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ be the vectors chosen randomly from Π_S to construct Π, i.e., $\Pi = \langle \boldsymbol{v}_1, \ldots, \boldsymbol{v}_m \rangle$. Then construct the sequence of subspaces $\Pi(i)$, $i = 0, \ldots, m$, as follows. First, set $\Pi(0) \triangleq \Pi_k$ and then define $\Pi(i)$ for $i \neq 0$ recursively, $\Pi(i) = \Pi(i-1) + \langle \boldsymbol{v}_i \rangle$. We also define $d(i) \triangleq \dim(\Pi(i))$, $i = 0, \ldots, m$. From Lemma 2.9, by choosing $\Pi_1 = \Pi_S$, $\Pi_2 = \Pi(i-1)$ and $m = 1$ we deduce that $d(i) = d(i-1) + 1$ with probability $1 - O\left(q^{-1}\right)$, unless $d(i-1) = n$.

Now we consider two cases. First, if $m + k \leq n$ then we have $\dim(\Pi + \Pi_k) = k + m$ or equivalently $\dim(\Pi \cap \Pi_k) = 0$ with high probability, i.e., $1 - O\left(q^{-1}\right)$. Secondly, when $m + k > n$ we have $\dim(\Pi + \Pi_k) = n$ with probability $1 - O\left(q^{-1}\right)$. From Lemma 2.8 we have $\dim(\Pi) = \min[m, n]$ w.h.p. So we have $\dim(\Pi \cap \Pi_k) = \dim(\Pi_k) + \dim(\Pi) - \dim(\Pi_k + \Pi) = k + \min[m, n] - n$.

Combining these two cases we can write

$$\dim(\Pi \cap \Pi_k) = \left(k + \min[m, n] - n\right)^+, \tag{2.43}$$

w.h.p., which completes the proof. □

Proof of Corollary 2.1. Let us define $\Pi_{12} = \Pi_1 \cap \Pi_2$, where $d_{12} = \dim(\Pi_{12})$. Using Lemma 2.10, and taking $\Pi_S = \Pi_1$ and $\Pi_k = \Pi_{12}$, we have

$$\dim(\hat{\Pi}_1 \cap \Pi_{12}) = \min\left[d_{12}, (m_1 - (d_1 - d_{12}))^+\right], \tag{2.44}$$

with probability $1 - O\left(q^{-1}\right)$. Now, we can write

$$\begin{aligned}\mathbb{P}\left[\hat{d}_{12} = \alpha\right] = \mathbb{P}\left[\hat{d}_{12} = \alpha | \dim(\hat{\Pi}_1 \cap \Pi_{12}) = \beta\right]\mathbb{P}\left[\dim(\hat{\Pi}_1 \cap \Pi_{12}) = \beta\right] \\ + \mathbb{P}\left[\hat{d}_{12} = \alpha | \dim(\hat{\Pi}_1 \cap \Pi_{12}) \neq \beta\right]\mathbb{P}\left[\dim(\hat{\Pi}_1 \cap \Pi_{12}) \neq \beta\right],\end{aligned} \tag{2.45}$$

where $\hat{d}_{12} = \dim(\hat{\Pi}_1 \cap \hat{\Pi}_2)$. Substituting $\beta = \min\left[d_{12}, (m_1 - (d_1 - d_{12}))^+\right]$ we obtain

$$\begin{aligned}\mathbb{P}\left[\hat{d}_{12} = \alpha\right] = \\ \mathbb{P}\left[\hat{d}_{12} = \alpha | \dim(\hat{\Pi}_1 \cap \Pi_{12}) = \beta\right]\left(1 - O\left(q^{-1}\right)\right) + O\left(q^{-1}\right).\end{aligned} \tag{2.46}$$

Selecting α properly and using Lemma 2.10 one more time, we get

$$\mathbb{P}\left[\hat{d}_{12} = \alpha\right] = 1 - O\left(q^{-1}\right), \tag{2.47}$$

where $\alpha = \min\left[\beta, (m_2 - (d_2 - \beta))^+\right]$, which completes the proof. □

Proof of Lemma 2.11. The results stated in the lemma follow from Corollary 2.1 by using induction on the number of subspaces. □

Proof of Theorem 2.3. To prove the theorem, it is sufficient to show that (2.38) is valid for one specific i with high probability. This is sufficient because if p_i is the probability that Π is in general position with respect to each Π_i, $i = 1, \ldots, r$, then the probability that Π is in general position with the whole family is lower bounded by $1 - \sum_{i=1}^{r}(1 - p_i)$.

Now by applying Lemma 2.10, we know that $p_i = 1 - O\left(q^{-1}\right)$ which completes the proof. □

Part I

Reducing Network Coding Overhead

Overview

There has been a growing consensus in the research community that *randomized* NC [4] is a promising technique to be applied in networking applications, such as wireless networks and content distribution networks. Due to its randomized nature and because practical networks are subjected to random delays, synchronization errors, packet erasures, nodes failures, and topology changes, it is not viable to assume that the linear combinations performed at the intermediate nodes are deterministically known at the receivers.

In practical networks, where such deterministic knowledge is not sustainable, the most popular approach is to append coding vectors at the headers of the packets to keep track of the linear combinations of the source packets they contain (see, e.g., [20]). This results in a loss of information rate that can be significant with respect to the min-cut value. In this scheme use of coding vectors is akin to use of training symbols to learn the transformation induced by a network.

An alternative approach is to assume a non-coherent scenario for communication, as proposed in [1], where neither the source(s) nor the receiver(s) have any knowledge of the network topology or the network nodes operations.

In this part, we focus on such a non-coherent communication scenario for a linear NC system where the following questions will be investigated:

1. What are the fundamental limits on the information transmission rates in a non-coherent NC scenario?
2. What kind of coding schemes can achieve the optimal rates?
3. How do the optimal rates compare to the coherent case and to the schemes using coding vectors?
4. Is it possible to reduce the overhead of NC in networks where the intermediate nodes operations are unknown.

To this end, we propose two different models where in both case the non-coherent NC channel is modeled by a multiplicative matrix channel over some finite field $\mathbb{F}_q$. In Chapter 3, we model the non-coherent NC by imposing a uniform and i.i.d. distribution over the channel transfer matrix in every time-slot. For this model we characterize the point-to-point capacity as well as the multiple sources (multiple access) rate region. On the other hand, in Chapter 4, we consider a partially known statistical model for the channel transfer

matrix, i.e., it is assumed that the transfer matrices changes independently and arbitrarily from time-slot to time-slot such that their rank distributions follow a known and fixed distribution. For both of these models we show that the subspace coding (originally proposed in [1]) is indeed information theoretically optimal to achieve the channel capacity. We further show that for the uniform transfer matrix model, the rate loss due to using coding vectors is of order $o_q(1)$ as the field size grows.

The result of Chapter 3 shows that in general it is not possible to further reduce the overhead of coding vectors because all of the possible linear combinations of the source packets are realizable. In contrast, in Chapter 5, we consider the problem of reducing the NC overhead for the case where the coding vectors are a sparse linear combinations of the source packets. Then, we present a novel scheme to reduce the coding vectors overhead without changing the operation of relay nodes in the network.

"The capacity to be puzzled is the premise of all creation, be it in art or in science."

- Erich Fromm

Capacity of Non-coherent Network Coding

3

The first fundamental result proved in NC, and perhaps still the most useful from a practical point of view today, is that, using linear NC [5, 7], one can achieve rates up to the common min-cut value when multicasting to $N_r \geq 1$ receivers. In general this may require operations over a field of size approximately $\sqrt{N_r}$, which translates to communication using packets of length $\frac{1}{2}\log N_r$ bits [21].

However, this result assumes that the receivers know perfectly the operations that the network nodes perform. In large dynamically changing networks, collecting network information comes at a cost, as it consumes bandwidth that could instead have been used for information transfer. In practical networks, where such deterministic knowledge is not sustainable, the most popular approach is to perform randomized NC [4] and to append coding vectors at the headers of the packets to keep track of the linear combinations of the source packets they contain (see, e.g., [20]). The coding vectors have an overhead of $h \log N_r$ bits, where h is the total number of packets to be linearly combined. This results in a loss of information rate that can be significant with respect to the min-cut value. In particular, in wireless networks such as sensor networks where communication is restricted to short packet lengths, the coding vector overhead can be a significant fraction of the overall packet length [22, 23].

Use of coding vectors is akin to use of training symbols to learn the transformation induced by a network. A different approach is to assume a non-coherent scenario for communication, as proposed in [1], where neither the source(s) nor the receiver(s) have any knowledge of the network topology or the network nodes operations. Non-coherent communication allows for creating end-to-end systems completely oblivious to the network state. Several natural questions arise considering this non-coherent framework: (i) what are the fundamental limits on the rates that can be achieved in a network where the intermediate

node operations are unknown, (ii) how can they be achieved, and (iii) how do they compare to the coherent case.

In this chapter we address such questions for two different cases. First, we consider the scenario where a single source aims to transmit information to one or multiple receiver(s) over a network under the non-coherence assumption using fixed packet length. Because network nodes only perform linear operations, the overall network behavior from the source(s) to a receiver can be represented as a matrix multiplication of the sent source packets. We consider operation in time-slots, and assume that the channel transfer matrices are distributed uniformly at random and i.i.d. over different time-slots. Under this probabilistic model, we characterize the asymptotic capacity behavior of the introduced channel and show that using *subspace coding* we can achieve the optimal performance. We extend our model for the case of multiple sources and characterize the asymptotic behavior of the optimal rate region for the case of two sources. We believe that this result can be extended to the case of more than two sources using the same method that is applied in Section 3.4. For the multi-source case we prove as well that encoding information using subspaces is sufficient to achieve the optimal rate region.

The idea of non-coherent modeling for randomized NC was first proposed in the seminal work by Koetter and Kschischang in [1]. In that work, the authors focused on algebraic subspace code constructions over a Grassmannian. Independently and in parallel to our work in [24], Montanari et al. [25] introduced a different probabilistic model to capture the end-to-end functionality of non-coherent NC operation, with a focus on the case of error correction capabilities. Their model does not examine subsequent time slots, but instead, allows the packets block length (in this chapter terminology; packet length L) to increases to infinity, with the result that the overhead of coding vectors becomes negligible, very fast.

Silva et al. [26] independently and subsequent to our works in [24] and [27], also considered a probabilistic model for non-coherent NC, which is an extension of the model introduced in [25] over multiple time-slots. In their model the transfer matrix is constrained to be square as well as full rank. This is in contrast to our model, where the transfer matrix can have arbitrary dimensions, and the elements of the transfer matrix are chosen uniformly at random, with the result that the transfer matrix itself may not have full rank (this becomes more pronounced for small matrices). Moreover, we extend our work to multiple source multicast, which corresponds to a virtual non-coherent MAC. Our results coincide for the case of a single source, when the packet length and the finite field of operations are allowed to grow sufficiently large. Another difference is that the work in [26] focuses on additive error with constant dimensions; in contrast, we focus on packet erasures.

Subsequent to our works [24, 27], Yang et al. [28, 29] (see also [30, 31]) considered a completely general scenario, making no assumption on the distribution of the transfer matrix. They obtained upper and lower bounds on the channel capacity, and give a sufficient condition on the distribution of the transfer matrix such that coding over subspaces is capacity achieving. They

also studied the achievable rates of coding over subspaces.

Nobrega et al. [32] considered the case where the probability distribution of the rank of the transfer matrix is arbitrary; however all matrices with the same rank are equiprobable. Then, following an approach similar to Chapter 3 (see also [33]), they expressed the capacity as the solution of a convex optimization problem over $O(\min[M, N])$ variables. They also observed that in this case the subspace codes are sufficient to achieve the capacity.

An interpretation of our results is that it is the finite field analog of the Grassmannian packing result for non-coherent MIMO channels as studied in the well known work in [34]. In particular, we show that for the non-coherent model over finite fields, the capacity critically depends on the relationship between the "coherence time" (or packet length L in our model) and the min-cut of the network. In fact the number of active subspace dimensions depend on this relationship; departing from the non-coherent MIMO analogy of [34].

All the missing proofs for lemmas, theorems, and etc., are given in Appendix 3.A unless otherwise stated.

It is important to mention that this chapter has been done as a joint work with Soheil Mohajer [1].

3.1 Channel Model and Notation

3.1.1 Notation

In this chapter we use the notation introduced in Section 2.1. Moreover, we use the calligraphic symbols, i.e., $\mathcal{X}$ or $\mathcal{Y}$ to denote a set of matrices. To denote a set of subspaces we use the same calligraphic symbols but with a "$\sim$", i.e., $\widetilde{\mathcal{X}}$ or $\widetilde{\mathcal{Y}}$.

For two real valued functions $f(x)$ and $g(x)$ of x, we use $f(x) \doteq g(x)$ to denote that [2]

$$\lim_{x\to\infty} \frac{\log f(x)}{\log g(x)} \to 1. \tag{3.1}$$

We also use a similar definition for $f \dot{\leq} g$ to denote that

$$\lim_{x\to\infty} \frac{\log f(x)}{\log g(x)} \to c \leq 1, \tag{3.2}$$

where c is a constant.

1. Soheil Mohajer was a Ph.D. student at Ecole Polytechnique Fédérale de Lausanne (EPFL), working under the supervision of prof. Suhas Diggavi. Now, he is doing a postdoc at U.C. Berkeley.

2. One has to specify the growing variable whenever "$\doteq$" is used for multi-variate functions. However, since in this work the growing variable is always q, the field size, we will not repeat it for sake of brevity.

3.1.2 The Non-Coherent Finite Field Channel Model

We consider a network where nodes perform random linear NC over a finite field $\mathbb{F}_q$. We are interested in the maximum information rate at which a single (or multiple) source(s) can successfully communicate over such a network when neither the transmitter nor the receiver(s) have any channel state information (CSI). For simplicity, we will present the channel model and our analysis for the case of a single receiver; the extension to multiple receivers (with the same channel parameters) is straightforward, as we also discuss in the results section.

We assume that time is slotted and the channel is block time-varying. For the single source communication, at time slot t, the receiver observes

$$\boldsymbol{Y}[t] = \boldsymbol{H}[t]\boldsymbol{X}[t], \tag{3.3}$$

where $\boldsymbol{X}[t] \in \mathbb{F}_q^{M\times L}$, $\boldsymbol{H}[t] \in \mathbb{F}_q^{N\times M}$, and $\boldsymbol{Y}[t] \in \mathbb{F}_q^{N\times L}$. At each time-slot, the receiver receives N packets of length L (captured by the rows of matrix $\boldsymbol{Y}[t]$) that are random linear combinations of the M packets injected by the source (captured by the rows of matrix $\boldsymbol{X}[t]$). In our model, the packet length L can be interpreted as the coherence time of the channel, during which the transfer matrix remains constant. Each element of the transfer matrix $\boldsymbol{H}[t]$ is chosen uniformly at random from $\mathbb{F}_q$, changes independently from time slot to time slot, and is unknown to both the source and the receiver. In other words, $\boldsymbol{H}[t] \sim \mathsf{Uni}\left(\mathbb{F}_q^{N\times M}\right)$ and has i.i.d. distribution over different blocks. In general, the topology of the network may impose some constraints on the transfer matrix $\boldsymbol{H}[t]$ (for example, some entries might be zero, see [7, 35, 36, 37]). However, we believe that this is a reasonable general model, especially for large-scale dynamically-changing networks where apart from random coefficients there exist many other sources of randomness. Formally, we define the non-coherent matrix channel as follows.

Definition 3.1 (Non-coherent matrix channel Ch_m)**.** *This is defined to be the matrix channel* $\mathrm{Ch}_m : \mathcal{X} \to \mathcal{Y}$ *described by (3.3) with the assumption that* $\boldsymbol{H}[t]$ *is i.i.d. and* $\boldsymbol{H}[t] \sim \mathsf{Uni}\left(\mathbb{F}_q^{N\times M}\right)$*. It is a discrete memory-less channel with input alphabet* $\mathcal{X} \triangleq \mathbb{F}_q^{M\times L}$ *and output alphabet* $\mathcal{Y} \triangleq \mathbb{F}_q^{N\times L}$*.*

The capacity of the channel Ch_m is given by

$$C_m = \max_{P_{\boldsymbol{X}}} I(\boldsymbol{X};\boldsymbol{Y}), \tag{3.4}$$

where $P_{\boldsymbol{X}}$ is the input distribution. To achieve the capacity a coding scheme may employ the channel given in (3.3) multiple times, and a codeword is a sequence of input matrices from $\mathcal{X}$. For a coding strategy that induces an input distribution $P_{\boldsymbol{X}}$, the achievable rate is

$$\mathfrak{R} = I(\boldsymbol{X};\boldsymbol{Y}). \tag{3.5}$$

Now we define a non-coherent subspace channel Ch_s which takes as an input a subspace and outputs another subspace. Then, in Theorem 3.1 we will show

that the two channels Ch_m and Ch_s are equivalent from the point of view of calculating the mutual information between their inputs and their outputs.

Definition 3.2 (Non-coherent subspace channel Ch_s)**.** *This is defined to be the channel* $\mathrm{Ch}_s : \widetilde{\mathcal{X}} \to \widetilde{\mathcal{Y}}$ *with input alphabet* $\widetilde{\mathcal{X}} = \mathrm{Sp}(L, M)$ *and output alphabet* $\widetilde{\mathcal{Y}} = \mathrm{Sp}(L, N)$ *and transition probability*

$$P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x) \triangleq \begin{cases} \psi(T, N, \pi_y) q^{-N \dim(\pi_x)} & \pi_y \sqsubseteq \pi_x, \\ 0 & \text{otherwise}, \end{cases} \tag{3.6}$$

where Π_X *and* Π_Y *are the input and output variables of the channel* Ch_s*.*

The capacity of the channel Ch_s is given by

$$C_s = \max_{P_{\Pi_X}} I(\Pi_X; \Pi_Y), \tag{3.7}$$

where P_{Π_X} is the input distribution defined over the set of subspaces $\widetilde{\mathcal{X}}$.

We next consider a multiple sources scenario, and the MAC corresponding to (3.3). In this case, we have

$$\boldsymbol{Y}[t] = \sum_{i=1}^{s} \boldsymbol{H}_i[t] \boldsymbol{X}_i[t], \tag{3.8}$$

where s is the number of sources, and each source i inserts M_i packets to the network. Thus, $\boldsymbol{X}_i[t] \in \mathbb{F}_q^{M_i \times L}$, $\boldsymbol{H}_i[t] \in \mathbb{F}_q^{N \times M_i}$ and $\boldsymbol{Y}[t] \in \mathbb{F}_q^{N \times L}$. We can also collect all $\boldsymbol{H}_i[t]$ in an $N \times \sum_{i=1}^{s} M_i$ matrix $\boldsymbol{H}_{\mathsf{MAC}}[t]$ and all $\boldsymbol{X}_i[t]$ in an $\sum_{i=1}^{s} M_i \times L$ matrix $\boldsymbol{X}_{\mathsf{MAC}}[t]$ as following

$$\boldsymbol{X}_{\mathsf{MAC}}[t] \triangleq \begin{bmatrix} \boldsymbol{X}_1[t] \\ \vdots \\ \boldsymbol{X}_s[t] \end{bmatrix}, \tag{3.9}$$

and

$$\boldsymbol{H}_{\mathsf{MAC}}[t] \triangleq \begin{bmatrix} \boldsymbol{H}_1[t] & \cdots & \boldsymbol{H}_s[t] \end{bmatrix}, \tag{3.10}$$

so we can rewrite (3.8) as

$$\boldsymbol{Y}[t] = \boldsymbol{H}_{\mathsf{MAC}}[t] \boldsymbol{X}_{\mathsf{MAC}}[t]. \tag{3.11}$$

Each source i then controls M_i rows of the matrix $\boldsymbol{X}_{\mathsf{MAC}}[t]$. Again we assume that each entry of the matrices $\boldsymbol{H}_i[t]$ is chosen i.i.d. and uniformly at random from the field $\mathbb{F}_q$ for all source nodes and all time instances.

Definition 3.3 (The non-coherent multiple access matrix channel $\mathrm{Ch}_{m-\mathsf{MAC}}$)**.** *This is defined to be the channel* $\mathrm{Ch}_{m-\mathsf{MAC}} : \mathcal{X}_1 \times \cdots \times \mathcal{X}_s \to \mathcal{Y}$ *described in* (3.8)*, with the assumption that* $\boldsymbol{H}_i[t]$*,* $i = 1, \ldots, s$*, are i.i.d. and uniformly distributed over all matrices* $\mathbb{F}_q^{N \times M_i}$*,* $i = 1, \ldots, s$*. It forms a discrete memoryless MAC with input alphabets* $\mathcal{X}_i \triangleq \mathbb{F}_q^{M_i \times L}$*,* $i = 1, \ldots, s$*, and output alphabet* $\mathcal{Y} \triangleq \mathbb{F}_q^{N \times L}$*.*

It is well known [38] that the rate region of any multiple access channel including $\mathrm{Ch}_{m-\mathsf{MAC}}$ is given by the closure of the convex hull of the rate vectors satisfying

$$\mathfrak{R}_{\mathcal{S}} \leq I(X_{\mathcal{S}}; Y | X_{\mathcal{S}^c}) \quad \forall \mathcal{S} \subseteq \{1, \ldots, s\}, \tag{3.12}$$

for some product distribution $P_{X_1}(x_1) \cdots P_{X_s}(x_s)$. Note that $\mathfrak{R}_{\mathcal{S}} = \sum_{i \in \mathcal{S}} \mathfrak{R}_i$ where $\mathfrak{R}_i$ is the transmission rate of the ith source, $X_{\mathcal{S}} = \{X_i : i \in \mathcal{S}\}$ and $\mathcal{S}^c$ is the complement set of $\mathcal{S}$.

As before, we define a non-coherent subspace version [3] of the matrix multiple access channel and in Theorem 3.6 we show that from the point of view of rate region these two channels are equivalent.

Definition 3.4 (Non-coherent subspace multiple access channel $\mathrm{Ch}_{s-\mathsf{MAC}}$)**.** *This is defined to be the channel* $\mathrm{Ch}_{s-\mathsf{MAC}} : \widetilde{\mathcal{X}}_1 \times \widetilde{\mathcal{X}}_2 \to \widetilde{\mathcal{Y}}$ *with input alphabets* $\widetilde{\mathcal{X}}_i = \mathrm{Sp}(L, M_i)$*,* $i = 1, 2$*, output alphabet* $\widetilde{\mathcal{Y}} = \mathrm{Sp}(L, N)$ *and transition probability*

$$P_{\Pi_Y|\Pi_{X_1},\Pi_{X_2}}(\pi_y|\pi_1, \pi_2) = \begin{cases} \psi(L, N, \pi_y) q^{-N \dim(\pi_1+\pi_2)} & \pi_y \sqsubseteq \pi_1 + \pi_2, \\ 0 & \text{otherwise}, \end{cases} \tag{3.13}$$

where Π_{X_1} *and* Π_{X_2} *are the input and* Π_Y *is the output variables of the channel* $\mathrm{Ch}_{s-\mathsf{MAC}}$*.*

3.2 Main Results

Here, in this section we state the main results of this chapter.

3.2.1 Single Source

Our main results, Theorem 3.2 and Theorem 3.3, characterize the capacity for non-coherent NC for the model given in (3.3). We show that the capacity is achieved through subspace coding, where the information is communicated from the source to the receivers through the choice of subspaces. Formally, we have the following results.

Theorem 3.1. *The matrix channel* $\mathrm{Ch}_m : \mathcal{X} \to \mathcal{Y}$ *defined in Definition 3.1 and the subspace channel* $\mathrm{Ch}_s : \widetilde{\mathcal{X}} \to \widetilde{\mathcal{Y}}$ *defined in Definition 3.2 are equivalent in terms of evaluating the mutual information between the input and output. More precisely, for every input distribution for the channel* Ch_s *there is an input distribution for the channel* Ch_m *such that* $I(\boldsymbol{X}; \boldsymbol{Y}) = I(\Pi_X; \Pi_Y)$ *and vice versa. As a result, these channels have the same capacity* $C_m = C_s$*.*

3. For simplicity, we restrict this definition to only two source nodes. However, generalization to s sources is straightforward.

For the proof of Theorem 3.1 refer to Appendix 3.A and for more discussion refer to Section 3.3.1.

Theorem 3.2. *For the channel* $\mathrm{Ch}_m : \mathcal{X} \to \mathcal{Y}$ *defined in Definition 3.1, the capacity is given by*

$$C_m = i^*(L - i^*) \log q + o(1), \tag{3.14}$$

where $i^* = \min\left[M, N, \lfloor \frac{L}{2} \rfloor\right]$, *and* $o(1)$ *tends to zero as* q *grows.*

Theorem 3.2 is proved in Section 3.3.2. The result of Theorem 3.2 is for large alphabet regime [4]. The following result, Theorem 3.3, is valid for a finite field size, and therefore is a non-asymptotic result.

Theorem 3.3. *Consider the channel* $\mathrm{Ch}_m : \mathcal{X} \to \mathcal{Y}$ *defined in Definition 3.1. There exists a finite number* q_0 *such that for* $q > q_0$ *the optimal input distribution is non-zero only for matrices of rank in the set*

$$\mathcal{A} = \left\{\min\left[(L-N)^+, M, N, L\right], \ldots, \min\left[M, N, L\right]\right\}. \tag{3.15}$$

Moreover, for all values of q *the optimal input distribution is uniform over all matrices* $\boldsymbol{X}$ *of the same rank, and the total probability allocated to transmitting matrices of rank* i *equals*

$$\alpha_i^* \triangleq \mathbb{P}\left[\mathrm{rank}(\boldsymbol{X}) = i\right] = 2^{-C_m} q^{i(L-i)} \left[1 + o(1)\right], \quad \forall i \in \mathcal{A}. \tag{3.16}$$

The proof of Theorem 3.3 is presented in Section 3.3.3 and Section 3.3.4, and uses standard techniques from convex optimization, as well as large field size approximations. Note that, the same coding scheme at the source simultaneously achieves the capacity for all receivers with the same channel parameters (i.e., values of M, N and L). That is, each receiver is able to successfully decode.

The result of Theorem 3.3 for the active set of input dimensions is not asymptotic in q. However, it is not easy to analytically find the minimum value of q_0 such that the theorem statement holds for all $q > q_0$. Theorem 3.4 demonstrates how we can analytically characterize q_0 given in Theorem 3.3 for the case $L > N + \min[M, N]$. The proof of Theorem 3.4 is presented in Section 3.3.5.

Theorem 3.4. *If* $L > N + \min[M, N]$, *then the capacity of* Ch_m *for* $q \geq q_0$ *is given by*

$$\begin{aligned} C_m &= \sum_{l=0}^{i^*} \psi(N, l) \begin{bmatrix} i^* \\ l \end{bmatrix} q^{-Ni^*} \log \left(\frac{\begin{bmatrix} \lceil L \rceil \\ l \end{bmatrix}}{\begin{bmatrix} i^* \\ l \end{bmatrix}} \right) \\ &= i^*(L - i^*) \log q - \mathbb{1}_{\{N \leq M\}}(L - i^*) \frac{\log q}{q} + q^{-1} + o(q^{-1}), \end{aligned} \tag{3.17}$$

4. We gratefully acknowledge the contribution of an anonymous reviewer who gave an alternate proof, which focused on the asymptotic q regime. We have included that proof in Section 3.3.2. Our original proof was based partially on the proof now given for Theorem 3.3.

where $\mathbb{1}_{\{.\}}$ *is the indicator function and* q_0 *is the minimum field size that satisfies the set of inequalities*

$$\frac{\epsilon_{q_0}(l) - \epsilon_{q_0}(i^*)}{(L-N-i^*)(i^*-l)} \leq \log q_0, \quad \forall l : 0 \leq l \leq (i^*-1), \tag{3.18}$$

and

$$\frac{\epsilon_{q_0}(l) - \epsilon_{q_0}(i^*)}{i^*(l-i^*)} \leq \log q_0, \quad \forall l : (i^*+1) \leq l \leq m, \tag{3.19}$$

where $i^* = \min[M, N]$ *and*

$$\epsilon_q(l) \triangleq \sum_{d_y=0}^{\min[N,l]} \psi(N, d_y) \begin{bmatrix} l \\ d_y \end{bmatrix} q^{-Nl} \log \left(\frac{\begin{bmatrix} L \\ d_y \end{bmatrix}}{\begin{bmatrix} i^* \\ d_y \end{bmatrix}} \right) - \min[N,l](L-i^*) \log q. \tag{3.20}$$

The capacity is achieved by sending matrices $\boldsymbol{X}$ *such that their rows span different* i^**-dimensional subspaces.*

Moreover, asymptotically in L*, we can show that* $q_0^{N-M+1} \geq 5M^2$ *is sufficient for the case* $M \leq N$ *and* $q_0 \geq NL$ *is sufficient if* $M > N$*.*

Theorems 3.2 and 3.3 state that the capacity behaves as $i^*(L-i^*)\log q$, for sufficiently large q. However, numerical simulations indicate a very fast convergence to this value as q increases. Figure 3.1 depicts the capacity for small values of q, calculated using the *Differential Evolution* toolbox for MATLAB [39]. This shows that the result is relevant at much lower field size than dictated by the formalism of the statement of Theorems 3.2 and 3.3.

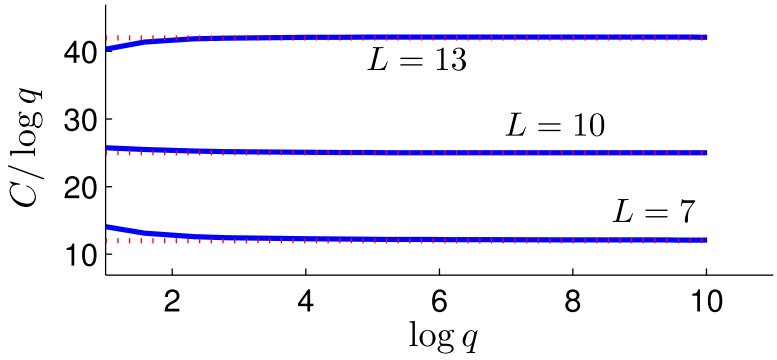


Figure 3.1 – Numerical calculation of the capacity for small values of q and $M = 11$, $N = 7$. The dotted line depicts $i^*(L-i^*)$.

From Theorem 3.3, we can derive the following guidelines for non-coherent network code design.

3.2.1.1 Choice of Subspaces

The optimal input distribution uses subspaces of a single dimension equal to $\min[M, N]$ for $L \geq \min[M, N] + N$. As L reduces, the set of used subspaces gradually increases, by activating one by one smaller and smaller dimensional subspaces, until, for $L \leq N$, all subspaces are used with equal probability [5]. Figure 3.2 pictorially depicts this gradual inclusion of subspaces.

This behavior is different from the result of [26] where all the subspaces up to dimension equal to the min-cut appeared in the optimal input distribution. This difference is due to the different channel model used in our work and in [26].

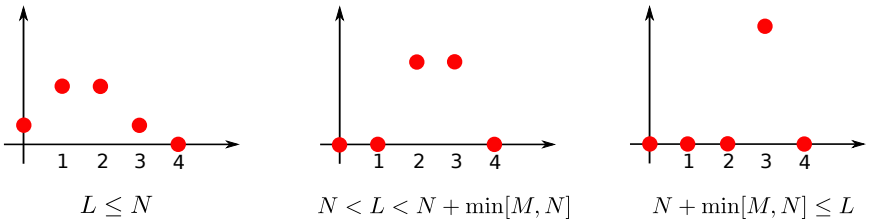


Figure 3.2 – Probability mass function of the active subspace dimensions for channel parameters $M = 4$, $N = 3$. As it is shown in Theorem 3.3 there exist three different regimes.

3.2.1.2 Values of M and N

For a given and fixed packet length L, the optimal value of M and N equals $M = N = \lfloor \frac{L}{2} \rfloor$ (optimality is in the sense of minimum requirement in order to obtain the maximum capacity for this L). For fixed L and M, the optimal value of N equals $N = \min[M, \lfloor \frac{L}{2} \rfloor]$. For fixed L and N, the optimal value of M equals $M = \min[N, \lfloor \frac{L}{2} \rfloor]$.

Table 3.1 – Information loss from using coding vectors when $N = M$.

	$L \leq 2M$	$L > 2M$
$C_m - \mathfrak{R}_{\text{cv}}$	$o(1)$	$o(1) = (i^* - 1)(L - i^*)\frac{\log q}{q} + O(q^{-1})$

5. Note that although all the subspaces are equiprobable, we have distinct values for α_i^* since there are different number of subspaces of each dimension.

3.2.1.3 Subspace Coding vs. Coding Vectors

One of the aims of this work was to find the regimes in which the using of coding vectors [20] is far from optimal. Table 3.1 summarizes this difference. As we see from the Table 3.1 subspace coding does not offer benefits as compared to the coding vectors approach for large field size [6].

Table 3.1 is calculated as follows. The achievable rate $\mathfrak{R}_{cv}$ using coding vectors equals

$$\mathfrak{R}_{cv} \triangleq \mathbb{P}\left[\text{rank}(\boldsymbol{H}_k) = k\right] \times k(L-k)\log q, \tag{3.21}$$

where $0 < k \leq M$ is the number of packets in each generation, i.e., each packet includes a coding vector of length k and $L-k$ information symbols. Equivalently, we assume that we use k out of the M possible input packets. The matrix $\boldsymbol{H}_k$ is the $k \times k$ sub-matrix of $\boldsymbol{H}$ that is applied over the input packets. To calculate $\mathfrak{R}_{cv}$, we know that

$$\mathbb{P}\left[\text{rank}(\boldsymbol{H}_k) = k\right] = \prod_{i=0}^{k-1}(1-q^{-k+i}) = 1 - q^{-1} + O(q^{-2}). \tag{3.22}$$

Assume we choose $k = i^*$, then we have

$$\mathfrak{R}_{cv} = i^*(L-i^*)\log q - i^*(L-i^*)\frac{\log q}{q}, \tag{3.23}$$

where $i^* = \min\left[M, N, \lfloor \frac{L}{2} \rfloor\right]$. For the capacity C_m we use the large q-regime as considered in Theorem 3.2 for the case $L \leq 2M$ and the finite q-regime of Theorem 3.4 for the case $L > 2M$.

3.2.2 Extension to the packet erasure networks

After the error free single source scenario, we consider packet erasure networks, and calculate an upper and lower bound on the capacity for this case. The work in [26], which is the closest to ours, did not consider erasures but instead constant-dimension additive errors. In practice, depending on the application, either of the models might be more suitable: for example, if NC is deployed at an application layer, then, unless there exist malicious attackers, packet erasures are typically used to abstract both the underlying physical channel errors, as well as packet dropped at queues or lost due to expired timers.

We model the erasures in the network as an end-to-end phenomenon which randomly erases packets according to some probability distribution. Formally, we rewrite the channel defined in (3.3) as

$$\boldsymbol{Y}[t] = \boldsymbol{E}[t]\boldsymbol{H}[t]\boldsymbol{X}[t], \tag{3.24}$$

6. In the algebraic framework of [1], the lifting construction used coding vectors, and they showed that this construction achieves almost the same rates as optimal algebraic subspace codes. However, we demonstrate in this chapter that this phenomenon occurs for longer packet lengths using an information-theoretic framework.

where $\boldsymbol{H}[t] \in \mathbb{F}_q^{M\times M}$ is assumed to be a square channel matrix and $\boldsymbol{E}[t] \in \mathbb{F}_q^{M\times M}$ is a diagonal random matrix whose elements on its diagonal are either 1 or 0. We also assume that q is large, and as a result the transfer matrix is full rank with high probability. Moreover, we consider the case where $M \leq \frac{L}{2}$, i.e. the matrix $\boldsymbol{X}[t]$ is a fat matrix. Recall that we can think of the rows of this matrix as packets send by the source, and the rows of the $\boldsymbol{Y}[t]$ matrix as packets received at the destination.

Note that in equation (3.24) all of the erasure events are captured by the erasure matrix $\boldsymbol{E}[t]$. Moreover, the erasure pattern is important only up to determining the number of packets that the destination receives, since the transfer matrix $\boldsymbol{H}[t]$ is unknown and distributed uniformly at random over all full rank matrices. Thus, we model the number of received packets (number of non-zero elements on the diagonal of $\boldsymbol{E}[t]$) as a random variable N (instead of a fixed N) which takes values in $0 \leq N \leq M$ according to some distribution that depends on the packet erasures in the network. In this case the capacity is

$$C_e = \max_{P_{\boldsymbol{X}}} I(\boldsymbol{X}; \boldsymbol{Y}, N). \tag{3.25}$$

We can then use our previous result, Theorem 3.2, to find an upper and lower bound for the capacity C_e when we have packet erasure in the network, as the following Theorem 3.5 describes.

Theorem 3.5. *Let the number of received packets at the destination be a random variable N defined over the set of integers $0 \leq N \leq M$. Also, assume that $M \leq \frac{L}{2}$. Then for large q, we have the following upper and lower bound for the capacity C_e,*

$$\mu_1(L-M)\log q \leq C_e \leq \mu_1\left(L - \frac{\mu_2}{\mu_1}\right)\log q, \tag{3.26}$$

where $\mu_1 \triangleq \mathbb{E}_N[N]$ and $\mu_2 \triangleq \mathbb{E}_N\left[N^2\right]$.

For the proof of Theorem 3.5 and more discussion refer to Appendix 3.B.

Remark 3.1. *Note that because we do not necessarily employ full-rank matrices $\boldsymbol{X}$, it is possible that although some packets are erased at the destination, the received packets still span a matrix of the same rank as $\boldsymbol{X}$; thus erasing packets is not equivalent to erasing dimensions.*

3.2.3 Multiple Sources

In several practical applications, such as sensor networks, data sources are not necessarily co-located. We thus extend our work to the case where multiple not co-located sources transmit information to a common receiver. In particular, we consider the non-coherent MAC introduced in Definition 3.3, and characterize the capacity region of this network for the case of two sources with M_1 and M_2 input packets and packet length $L > 2(M_1 + M_2)$. We believe that this technique can be extended to more than two sources.

To find the rate region of the matrix multiple access channel $\mathrm{Ch}_{m-\mathsf{MAC}}$, we first show that the two channels $\mathrm{Ch}_{m-\mathsf{MAC}}$ and $\mathrm{Ch}_{s-\mathsf{MAC}}$ are equivalent, as stated in Theorem 3.6. We then find the rate region of the subspace multiple access channel $\mathrm{Ch}_{s-\mathsf{MAC}}$ which is stated in Theorem 3.7. To avoid repetition, we state Theorem 3.6 without a proof because its proof is very similar to that of Theorem 3.1.

Theorem 3.6. *The matrix MAC channel* $\mathrm{Ch}_{m-\mathsf{MAC}}$ *defined in Definition 3.3 is equivalent to the subspace MAC channel* $\mathrm{Ch}_{s-\mathsf{MAC}}$ *defined in Definition 3.4 in the sense that the optimal rate region for these two channels is the same.*

Theorem 3.7. *For* $L > 2(M_1 + M_2)$*, the asymptotic (in the field size* q*) capacity region of the MAC* $\mathrm{Ch}_{m-\mathsf{MAC}}$ *introduced in Definition 3.3 is given by*

$$\mathcal{R}^* \triangleq \text{convex hull} \bigcup_{(d_1,d_2)\in\mathcal{D}^*} \mathcal{R}(d_1,d_2), \tag{3.27}$$

where

$$\mathcal{R}(d_1,d_2) \triangleq \big\{(\mathfrak{R}_1,\mathfrak{R}_2) : \mathfrak{R}_i \le \mathfrak{R}_i(d_1,d_2),\ i=1,2\big\}, \tag{3.28}$$

$$\mathfrak{R}_i(d_1,d_2) \triangleq d_i(L-d_1-d_2)\log q, \qquad i=1,2, \tag{3.29}$$

and

$$\mathcal{D}^* \triangleq \big\{(d_1,d_2):\ 0 \le d_i \le \min[N,M_i],\ 0 \le d_1+d_2 \le \min[N,M_1+M_2]\big\}. \tag{3.30}$$

We note that the rate region forms a polytopes that has the following number of corner points (see Corollary 3.1 in Section 3.4)

$$\min\left[M_1,(N-M_2)^+\right] + \min\left[M_2,(N-M_1)^+\right] + 2 - \mathbb{1}_{\{N\ge M_1+M_2\}}. \tag{3.31}$$

The rate region $\mathcal{R}^*$ is shown in Figure 3.3 for a particular choice of parameters.

The proof of this theorem is provided in Section 3.4. We first derive an outer bound by deriving two other bounds: a cooperative bound and a coloring bound. For the coloring bound, we utilize a combinatorial approach to bound the number of *distinguishable* symbol pairs that can be transmitted from the sources to the receiver. We then show that a simple scheme that uses coding vectors achieves the outer bound. We thus conclude that, for the case of two sources when $\frac{L}{2} > M_1 + M_2$, use of coding vectors is (asymptotically) optimal.

3.3 The Channel Capacity: Single Source Scenario

In this section we will prove Theorem 3.2, Theorem 3.3, and Theorem 3.4.

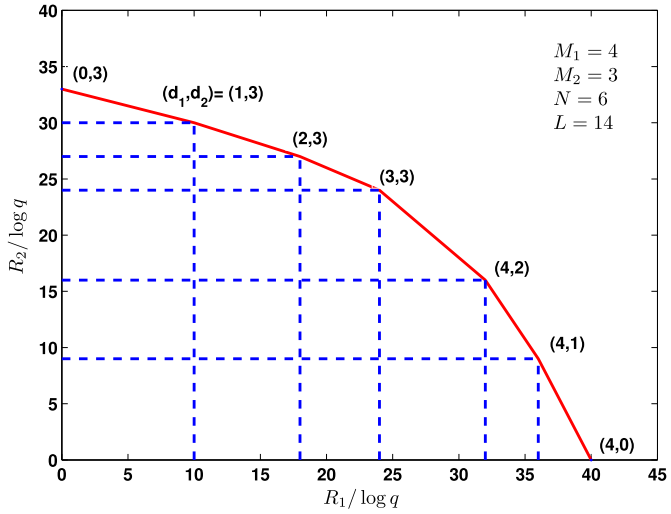


Figure 3.3 – The MAC region $\mathcal{R}^*$ for parameters $M_1 = 4$, $M_2 = 3$, $N = 6$, $L = 14$.

3.3.1 Equivalence of the Matrix Channel $\mathrm{Ch}_{\mathbf{m}}$ and the Subspace Channel $\mathrm{Ch}_{\mathbf{s}}$

For convenience let us rewrite the channel (3.3) again[7]

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X}. \tag{3.32}$$

To find the capacity of the above channel we need to maximize the mutual information between the input and the output of the channel with respect to the input distribution $P_{\boldsymbol{X}}$. Since the rows of $\boldsymbol{H}$ are chosen independently of each other, assuming that a matrix $\boldsymbol{X} = \boldsymbol{x}$ has been transmitted, we can think of the rows of the received matrix $\boldsymbol{Y}$ as chosen independently from each other, among all the possible vectors in the row span of $\boldsymbol{x}$. The independence of rows of $\boldsymbol{Y}$ allows us to write the conditional probability of $\boldsymbol{Y}$ given $\boldsymbol{X}$, referred to as the channel transition probability, as follows

$$P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x}) = \begin{cases} q^{-N\dim(\langle \boldsymbol{x}\rangle)} & \langle \boldsymbol{y}\rangle \sqsubseteq \langle \boldsymbol{x}\rangle\,, \\ 0 & \text{otherwise}, \end{cases} \tag{3.33}$$

where $\boldsymbol{x} \in \mathcal{X} = \mathbb{F}_q^{M\times L}$, and $\boldsymbol{y} \in \mathcal{Y} = \mathbb{F}_q^{N\times L}$.

7. In the rest of the chapter we will omit for convenience the time index t.

The mutual information $I(\boldsymbol{X};\boldsymbol{Y})$ between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is a function of $P_{\boldsymbol{X}}(\boldsymbol{x})$ and $P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})$ that can be expressed as

$$I(\boldsymbol{X};\boldsymbol{Y}) = \sum_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{y}\in\mathcal{Y}} P_{\boldsymbol{X}}(\boldsymbol{x})P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})\log\left(\frac{P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})}{P_{\boldsymbol{Y}}(\boldsymbol{y})}\right). \tag{3.34}$$

It is clear from (3.33) that $P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x}_1) = P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x}_2)$ for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X}$ such that $\langle \boldsymbol{x}_1\rangle = \langle \boldsymbol{x}_2\rangle$ which reveals symmetry for the channel Ch_m. We exploit this symmetry to show that $C_m = C_s$ as it is stated in Theorem 3.1 and proved in Appendix 3.A.

The proof of Theorem 3.1 determines how we can map an input distribution of Ch_s to an input distribution for Ch_m that achieves the same mutual information. The input distribution $P_{\boldsymbol{X}}(\boldsymbol{x})$ should be chosen such that we have

$$\sum_{\boldsymbol{x}\in\mathcal{X}:\langle\boldsymbol{x}\rangle=\pi_x} P_{\boldsymbol{X}}(\boldsymbol{x}) = P_{\Pi_X}(\pi_x). \tag{3.35}$$

One simple way to do this is to put all the probability mass of π_x on one matrix $\boldsymbol{x}$ such that $\langle \boldsymbol{x}\rangle = \pi_x$.

3.3.2 Upper and Lower bound for the Capacity of Ch_m

Here, we state the proof of Theorem 3.2 by giving upper and lower bounds for the capacity that differ in $o(1)$ bits, which vanishes as $q \to \infty$.

Let $C_m(N, M, L)$ denote the capacity of the channel Ch_m. Let $C_{\text{f-}m}(N, M, L)$ denote the capacity of the channel $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$ where $\boldsymbol{A} \in \mathbb{F}_q^{N\times M}$ is a full-rank matrix chosen uniformly at random among all the full-rank matrices in $\mathbb{F}_q^{N\times M}$. Then, we have the following lemma.

Lemma 3.1. *We can bound $C_m(N, M, L)$ from above and below as follows*

$$C_m(h, h, L) \le C_m(N, M, L) \le C_{\text{f-}m}(N, M, L) \le C_{\text{f-}m}(h, h, L), \tag{3.36}$$

where $h = \min[M, N]$.

Proof. Let $\boldsymbol{U}_{N\times M} \in \mathbb{F}_q^{N\times M}$ denote a generic random matrix chosen uniformly at random and independently from any other variable. Similarly, let $\boldsymbol{A}_{N\times M} \in \mathbb{F}_q^{N\times M}$ denote a generic *full-rank* matrix chosen uniformly at random among all such full-rank matrices and independent from any other variable. (Note that each new instance of such a matrix in the same equation denotes a different random variable which is independent from the other random variables.)

Since the channel $\boldsymbol{Y} = \boldsymbol{A}_{N\times M}\boldsymbol{X}$ is statistically equivalent to the channel $\boldsymbol{Y} = \boldsymbol{A}_{N\times N}\boldsymbol{A}_{N\times M}\boldsymbol{A}_{M\times M}\boldsymbol{X}$, we have, by the data processing inequality, that $C_{\text{f-}m}(N, M, L) \le C_{\text{f-}m}(h, h, L)$.

Using the same argument, since the channel $\boldsymbol{Y} = \boldsymbol{U}_{N\times M}\boldsymbol{X}$ is equivalent to the channel $\boldsymbol{Y} = \boldsymbol{U}_{N\times N}\boldsymbol{A}_{N\times M}\boldsymbol{X}$ if $N \ge M$, and is equivalent to the channel $\boldsymbol{Y} = \boldsymbol{A}_{N\times M}\boldsymbol{U}_{M\times M}\boldsymbol{X}$ if $N \le M$ we have $C_m(N, M, L) \le C_{\text{f-}m}(N, M, L)$.

To obtain the lower bound we proceed as follows. Let us choose $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{I}_h \\ \boldsymbol{0} \end{bmatrix} \overline{\boldsymbol{X}}$ and $\overline{\boldsymbol{Y}} = [\boldsymbol{I}_h \;\; \boldsymbol{0}]\, \boldsymbol{Y}$, where $\boldsymbol{Y} = \boldsymbol{U}_{N\times M}\boldsymbol{X}$. Then we can write

$$\overline{\boldsymbol{Y}} = [\boldsymbol{I}_h \;\; \boldsymbol{0}]\, \boldsymbol{U}_{N\times M} \begin{bmatrix} \boldsymbol{I}_h \\ \boldsymbol{0} \end{bmatrix} \overline{\boldsymbol{X}} = \boldsymbol{U}_{h\times h}\overline{\boldsymbol{X}}, \tag{3.37}$$

where $\boldsymbol{U}_{h\times h}$ is the upper left $h \times h$ sub-matrix of $\boldsymbol{U}_{N\times M}$. Thus, again the data processing inequality implies that $C_m(h,h,L) \le C_m(N,M,L)$. □

Lemma 3.2. *For $C_m(N,M,L)$ we have*

$$C_m(N,M,L) \le i^*(L-i^*)\log q + o(1), \tag{3.38}$$

where $i^* = \min[M,N,\lfloor \frac{L}{2} \rfloor]$.

Proof. By Lemma 3.1 we have

$$\begin{aligned} C_m(N,M,L) &\le C_{f\text{-}m}(h,h,L) \\ &\overset{(a)}{=} \log\left(\sum_{i=0}^{h} \begin{bmatrix} L \\ i \end{bmatrix}\right) \\ &\overset{(b)}{=} i^*(L-i^*)\log q + o(1), \end{aligned} \tag{3.39}$$

where (a) follows from [26, Corollary 2] and (b) follows from Lemma 2.1. □

Lemma 3.3. *For $C_m(N,M,L)$ we have*

$$C_m(N,M,L) \ge i^*(L-i^*)\log q - o(1), \tag{3.40}$$

where $i^* = \min[M,N,\lfloor \frac{L}{2} \rfloor]$.

Proof. For every subspace $\Pi \in \mathrm{Gr}(L,i^*)$, let $\mathrm{RREF}(\Pi) \in \mathbb{F}_q^{i^*\times L}$ be a matrix in reduced row echelon form such that $\Pi = \langle \mathrm{RREF}(\Pi)\rangle$. Choose $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{I}_{i^*} \\ \boldsymbol{0} \end{bmatrix} \times \mathrm{RREF}(\Pi_X) \in \mathbb{F}_q^{M\times L}$, where Π_X is chosen uniformly at random from $\mathrm{Gr}(L,i^*)$. Define the random variable $Q = \mathbb{1}_{\{\mathrm{rank}(\boldsymbol{Y})=i^*\}}$. Note that $\Pi_Y = \Pi_X$ when $Q=1$. Thus, we have $H(\Pi_Y|\Pi_X, Q=1) = 0$ and $H(\Pi_Y|Q=1) = H(\Pi_X) = \log \begin{bmatrix} L \\ i^* \end{bmatrix} \ge i^*(L-i^*)\log q$. Then, it follows that

$$\begin{aligned} C_m(N,M,L) &\overset{(a)}{\ge} C_m(h,h,L) \\ &\overset{(b)}{\ge} I(\Pi_X;\Pi_Y) \\ &\overset{(c)}{=} I(\Pi_X;\Pi_Y,Q) \\ &= I(\Pi_X;Q) + I(\Pi_X;\Pi_Y|Q) \\ &\ge \mathbb{P}[Q=1] I(\Pi_X;\Pi_Y|Q=1) \\ &\ge \mathbb{P}[Q=1] i^*(L-i^*)\log q, \end{aligned} \tag{3.41}$$

where (a) is due to Lemma 3.1, (b) follows follows from Theorem 3.1, and (c) holds since Q is a deterministic function of Π_Y. Now, note that we can write

$$\begin{aligned}\mathbb{P}[Q=1] &= \mathbb{P}[\text{rank}(\boldsymbol{U}_{h\times h}\boldsymbol{X}) = i^*] \\ &= \mathbb{P}\left[\text{rank}\left(\boldsymbol{U}_{h\times h}\begin{bmatrix}\boldsymbol{I}_{i^*}\\ \boldsymbol{0}\end{bmatrix}\right) = i^*\right] \\ &= \mathbb{P}[\text{rank}(\boldsymbol{U}_{h\times i^*}) = i^*] \\ &\geq 1 - \frac{i^*}{q^{k-i^*+1}} \\ &\geq 1 - \frac{i^*}{q}, \end{aligned} \tag{3.42}$$

and thus we obtain the desired result. □

Combining Lemma 3.2 and Lemma 3.3 recovers Theorem 3.2.

3.3.3 The Optimal Solution: General Approach

Generally, we are interested in finding the capacity and input distribution of Ch_m exactly. It is shown in Theorem 3.1 that instead of the channel Ch_m we can focus on the channel Ch_s. Thus, we are interested in optimizing the following quantity

$$I(\Pi_X;\Pi_Y) = \sum_{\substack{\pi_x\in\widetilde{\mathcal{X}},\\ \pi_y\in\widetilde{\mathcal{Y}}}} P_{\Pi_X}(\pi_x)P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x)\log\left(\frac{P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x)}{P_{\Pi_Y}(\pi_y)}\right). \tag{3.43}$$

Remember that $\widetilde{\mathcal{X}} = \text{Sp}(L,M)$ and $\widetilde{\mathcal{Y}} = \text{Sp}(L,N)$.

The following lemma states that the optimal solution for the channel Ch_s should be uniform over all subspaces with the same dimension, as it is intuitively expected from the symmetry of the channel.

Lemma 3.4. *The input distribution that maximizes $I(\Pi_X;\Pi_Y)$ for Ch_s is the one which is uniform over all subspaces having the same dimension.*

Lemma 3.4 shows that the optimal input distribution can be expressed as

$$\mathbb{P}[\Pi_X = \pi_x] = \frac{\alpha_{d_x}}{\begin{bmatrix}L\\ d_x\end{bmatrix}}, \tag{3.44}$$

where $d_x = \dim(\pi_x)$, $\alpha_{d_x} = \mathbb{P}[\dim(\Pi_X) = d_x]$, and we have $\sum_{d_x=0}^{\min[M,L]}\alpha_{d_x} = 1$. We can then simplify $I(\Pi_X;\Pi_Y)$ as stated in the following lemma.

Lemma 3.5. *Assuming an optimal input probability distribution of the form in (3.44), the mutual information $I(\Pi_X;\Pi_Y)$ can be simplified to*

$$\begin{aligned} I(\Pi_X;\Pi_Y) = &- \sum_{d_x=0}^{\min[M,L]} \alpha_{d_x} N d_x \log q \\ &- \sum_{d_x=0}^{\min[M,L]} \alpha_{d_x} q^{-N d_x} \sum_{d_y=0}^{\min[N,d_x]} \psi(N,d_y) \begin{bmatrix} d_x \\ d_y \end{bmatrix} \log(f(d_y)), \end{aligned} \tag{3.45}$$

where

$$f(d_y) \triangleq \frac{P_{\Pi_Y}(\pi_y)}{\psi(N,d_y)} = \frac{1}{\begin{bmatrix} L \\ d_y \end{bmatrix}} \sum_{d_x=d_y}^{\min[M,L]} \begin{bmatrix} d_x \\ d_y \end{bmatrix} q^{-N d_x} \alpha_{d_x}. \tag{3.46}$$

Lemmas 3.4 and 3.5 show that the problem of finding the optimal input distribution for the channel Ch_s is reduced to finding the optimal choice for $\alpha_i,\ i = 0,\ldots,\min[M,L]$. We know that the mutual information is a concave function with respect to $P_{\Pi_X}(\pi_x)$'s. Observation 3.1 implies that because (3.44) is a linear transformation from $P_{\Pi_X}(\pi_x)$'s to α_i's, as a result the mutual information $I(\Pi_X;\Pi_Y)$ is also concave with respect to α_i's [40].

Observation 3.1. *Let $g(\mathbf{x})$ be a concave function and let $\mathbf{x} = h(\mathbf{z})$ be a linear transform from $\mathbf{z}$ to $\mathbf{x}$. Then $g(h(\mathbf{z}))$ is also a concave function.*

Using Observation 3.1, we know that the mutual information is a concave function with respect to α_i's. This allows us to use the Kuhn-Tucker theorem [40] to solve the convex optimization problem. According to this theorem, the set of probabilities α_i^*, $0 \le i \le \min[M,L]$, maximize the mutual information if and only if there exists some constant λ such that

$$\begin{cases} \left.\frac{\partial I(\Pi_X;\Pi_Y)}{\partial \alpha_k}\right|_{\boldsymbol{\alpha}^*} = \lambda \quad \forall k:\ \alpha_k^* > 0, \\ \left.\frac{\partial I(\Pi_X;\Pi_Y)}{\partial \alpha_k}\right|_{\boldsymbol{\alpha}^*} \le \lambda \quad \forall k:\ \alpha_k^* = 0, \end{cases} \tag{3.47}$$

where $\sum_{i=0}^{\min[M,L]} \alpha_i^* = 1$, $0 \le k \le \min[M,L]$, and $\boldsymbol{\alpha}^*$ is the vector of the optimum input probabilities of choosing subspaces of certain dimension,

$$\boldsymbol{\alpha}^* = \begin{bmatrix} \alpha_0^* & \cdots & \alpha_{\min[M,L]}^* \end{bmatrix}^{\mathrm{T}}. \tag{3.48}$$

Lemma 3.6. *By taking the partial derivative of the mutual information given in (3.45) with respect to α_k, we have*

$$\begin{aligned} I'_k &\triangleq \frac{\partial I(\Pi_X;\Pi_Y)}{\partial \alpha_k} \\ &= -Nk\log q - \sum_{d_y=0}^{\min[N,k]} \psi(N,d_y) \begin{bmatrix} k \\ d_y \end{bmatrix} q^{-Nk} \log\left(f(d_y)\right) - \log e. \end{aligned} \tag{3.49}$$

Multiplying both sides of (3.49) by α_k and summing over k we get

$$I - \log e = \sum_{k=0}^{\min[M,L]} \alpha_k I'_k. \tag{3.50}$$

By choosing the optimal values $\alpha_k = \alpha_k^*$ for $0 \leq k \leq \min[m,T]$, the RHS becomes λ, and the mutual information increases to C_s. So we may write $\lambda = C_s - \log e$.

3.3.4 Solution for Large Field Size

In this subsection, we focus on large size fields, $q \gg 1$. This assumption allows us to use some approximations to simplify the conditions in (3.47). Assuming large q we can rewrite (3.49) as follows

$$\begin{aligned} I'_k = &-Nk \log q - \log e \\ &- \sum_{d_y=0}^{\min[N,k]} \left(1 + O(q^{-1})\right) q^{-(N-d_y)(k-d_y)} \log\left(f(d_y)\right), \end{aligned} \tag{3.51}$$

where we have used Lemma 2.1 and Lemma 2.6. Using similar approximations, $\log f(d_y)$ defined in (3.46) can be approximated as

$$\begin{aligned} \log\left(f(d_y)\right) = &-d_y L \log q + O(q^{-1}) \\ &+ \log\left(\sum_{d_x=d_y}^{\min[M,L]} q^{-(N-d_y)d_x} \alpha_{d_x}\right). \end{aligned} \tag{3.52}$$

Then we have the following result, Lemma 3.7.

Lemma 3.7. *The dominating term in the summation in (3.51) is the one obtained for* $d_y = \min[N,k]$.

From the proof of Lemma 3.7 written in Appendix 3.A, we can also see that the remaining terms in the summation of (3.51) are of order $o(1)$, so we can write

$$\begin{aligned} I'_k = &[L \min[N,k] - Nk] \log q + \underbrace{o(1)}_{\epsilon_q(k)} - \log e \\ &- \log\left(\sum_{d_x=\min[N,k]}^{\min[M,L]} q^{-[N-\min[N,k]]d_x} \alpha_{d_x}\right). \end{aligned} \tag{3.53}$$

Assuming that the expression inside the $\log(\cdot)$ function in (3.53) is not zero for every $0 \leq k \leq \min[M,L]$, we can rewrite the Kuhn-Tucker conditions as

$$\sum_{d_x=\min[N,k]}^{\min[M,L]} q^{-[N-\min[N,k]]d_x} \alpha_{d_x} \geq 2^{-C_s + o(1)} q^{[L\min[N,k] - Nk]}, \tag{3.54}$$

where the inequality holds with equality for all k with $\alpha_k^* > 0$.

Let $\delta \triangleq \min[M, L]$ and define the $(\delta+1) \times (\delta+1)$ matrix $\mathbf{A}$ with elements

$$\mathbf{A}_{ij} \triangleq \begin{cases} q^{-[N-\min[N,i]]j} & \min[N,i] \le j \le \delta, \\ 0 & \text{otherwise.} \end{cases} \tag{3.55}$$

We also define the column vector $\mathbf{b}$ with elements $b_i \triangleq q^{[L\min[N,i]-Ni]}$ for $0 \le i \le \delta$. Note that for convenience the indices of matrix $\mathbf{A}$ and vector $\mathbf{b}$ start from 0. Using these definitions, we are able to rewrite the Kuhn-Tucker conditions in the matrix form as

$$\mathbf{A}\boldsymbol{\alpha}^* \succeq 2^{-C_s+o(1)}\mathbf{b}. \tag{3.56}$$

In the following, we consider two cases for $\delta \le N$ and $\delta > N$, and find $\boldsymbol{\alpha}^*$ for each of them, separately.

First case: $\delta \le N$. In this case we can explicitly write the matrix $\mathbf{A}$ and vector $\mathbf{b}$ as

$$\mathbf{A} = \begin{bmatrix} 1 & q^{-N} & \cdots & q^{-(\delta-1)N} & q^{-\delta N} \\ 0 & q^{-(N-1)} & \cdots & q^{-(\delta-1)(N-1)} & q^{-\delta(N-1)} \\ 0 & 0 & \cdots & q^{-(\delta-1)(N-2)} & q^{-\delta(N-2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & q^{-(\delta-1)(N-\delta+1)} & q^{-\delta(N-\delta+1)} \\ 0 & 0 & \cdots & 0 & q^{-\delta(N-\delta)} \end{bmatrix}, \tag{3.57}$$

and

$$\mathbf{b} = \begin{bmatrix} 1 & q^{(L-N)} & \cdots & q^{\delta(L-N)} \end{bmatrix}^{\mathrm{T}}. \tag{3.58}$$

The fact that the expression inside the $\log(\cdot)$ function in (3.53) is non-zero for $k = \delta$, forces α_δ^* to be positive. Thus the last row of the matrix inequality in (3.56) should be satisfied as an equality. Therefore,

$$\alpha_\delta^* = \frac{q^{\delta(L-N)}}{q^{-\delta(N-\delta)}} 2^{-C_s+o(1)} = q^{\delta(L-\delta)} 2^{-C_s+o(1)}. \tag{3.59}$$

Now we use induction to show that the optimal solution has the form

$$\alpha_i^* = \begin{cases} q^{i(L-i)} 2^{-C_s+o(1)} & : \quad \kappa \le i \le \delta, \\ 0 & : \quad 0 \le i < \kappa, \end{cases} \tag{3.60}$$

where we will determine κ later.

Let us fix l and assume that $\alpha_i^* = q^{i(L-i)} 2^{-C_s+o(1)}$ for $0 \le l < i \le \delta$. Then for α_l^* we can write

$$A_{ll}\alpha_l^* + \sum_{j=l+1}^{\delta} q^{-(N-l)j} \alpha_j^* \ge q^{l(L-N)} 2^{-C_s+o(1)}, \tag{3.61}$$

or equivalently

$$\begin{aligned} A_{ll}\alpha_l^* \geq & q^{l(L-N)}2^{-C_s+o(1)} - \sum_{j=l+1}^{\delta} q^{-(N-l)j}\alpha_j^* \\ = & q^{l(L-N)}2^{-C_s+o(1)}\left[1 - \sum_{j=l+1}^{\delta} q^{(L-N-j)(j-l)}\right]. \end{aligned} \tag{3.62}$$

We can use induction for one step more to show that α_l^* is of the desired form (3.60) if the previous expression is satisfied with equality. This is true if we have $1 - \sum_{j=l+1}^{\delta} q^{(L-N-j)(j-l)} \geq 0$, or equivalently (assuming large q) if we have $(L-N-j)|_{j=l+1} < 0$. So we can conclude that we should have $(L-N)^+ \leq l \leq \delta$. It can be easily verified that for $i < (L-N)^+$ the Kuhn-Tucker equation for α_i^* satisfies the strict inequality so $\alpha_i^* = 0$ for $i < \min[(L-N)^+, \delta]$. The above argument results in a solution of the following form for the case $\delta \leq N$

$$\alpha_i^* = \begin{cases} q^{i(L-i)}2^{-C_s+o(1)} & : \quad \min\left[(L-N)^+, \delta\right] \leq i \leq \delta, \\ \\ 0 & : \quad 0 \leq i < \min\left[(L-N)^+, \delta\right]. \end{cases} \tag{3.63}$$

Second case: $\delta > N$. We now write matrix $\mathbf{A}$ and vector $\mathbf{b}$ as

$$\mathbf{A} = \left[\begin{array}{ccccccc} 1 & q^{-N} & \cdots & \cdots & \cdots & \cdots & q^{-\delta N} \\ 0 & q^{-(N-1)} & \cdots & \cdots & \cdots & \cdots & q^{-\delta(N-1)} \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & q^{-(N-1)} & q^{-N} & \cdots & q^{-\delta} \\ \hline 0 & \cdots & 0 & 0 & 1 & \cdots & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 1 & \cdots & 1 \end{array}\right], \tag{3.64}$$

and

$$\mathbf{b} = \begin{bmatrix} 1 & q^{(L-N)} & \cdots & q^{(N-1)(L-N)} & q^{N(L-N)} & q^{N(L-N-1)} & \cdots & q^{N(L-\delta)} \end{bmatrix}^{\mathrm{T}}. \tag{3.65}$$

The last $\delta - N + 1$ rows of $\mathbf{A}$ are the same while b_i is decreasing with i for $i \geq N$. Thus, the last $\delta - N$ inequalities are strict and therefore,

$$\alpha_{N+1}^* = \cdots = \alpha_\delta^* = 0. \tag{3.66}$$

The remaining equations can simply be reduced to the first case. Define

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & q^{-N} & \cdots & q^{-(N-1)N} & q^{-N^2} \\ 0 & q^{-(N-1)} & \cdots & q^{-(N-1)(N-1)} & q^{-N(N-1)} \\ 0 & 0 & \cdots & q^{-(N-1)(N-2)} & q^{-N(N-2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & q^{-(N-1)} & q^{-N} \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}, \tag{3.67}$$

and

$$\tilde{\mathbf{b}} = \begin{bmatrix} 1 & q^{(L-N)} & \cdots & q^{N(L-N)} \end{bmatrix}^{\mathrm{T}}. \tag{3.68}$$

The remaining conditions in this case can be written as

$$\tilde{\mathbf{A}}\boldsymbol{\alpha}^* \succeq 2^{-C_s+o(1)}\tilde{\mathbf{b}}, \tag{3.69}$$

which is exactly similar to (3.56), for $\delta = N$. Therefore, the optimal solution for the first case will also satisfy these conditions, i.e.,

$$\alpha_i^* = \begin{cases} q^{i(L-i)}2^{-C_s+o(1)} & \kappa \le i \le N, \\ 0 & 0 \le i < \kappa, \end{cases} \tag{3.70}$$

with $\kappa = \min[(L-N)^+, N]$. Summarizing (3.66) and (3.70), we can obtain the optimal solution for this regime, as

$$\alpha_i^* = \begin{cases} 0 & N < i \le \delta, \\ q^{i(L-i)}2^{-C_s+o(1)} & \kappa \le i \le N, \\ 0 & 0 \le i < \kappa, \end{cases} \tag{3.71}$$

where $\kappa = \min[(L-N)^+, N]$. This completes the proof of Theorem 3.3. By normalizing α_i^* to 1 we can also obtain an alternative proof to Theorem 3.2.

Remark 3.2. *To characterize the exact value of q_0 one have to consider the exact form of the set of equations given in* (3.62) *(for each l) which are as follows,*

$$A_{ll}\alpha_l^* \ge q^{l(L-N)}2^{-C_s+\epsilon_q(l)}\left[1 - \sum_{j=l+1}^{\delta} q^{(L-N-j)(j-l)}2^{[\epsilon_q(j)-\epsilon_q(l)]}\right], \tag{3.72}$$

where $\epsilon_q(\cdot)$ is defined in (3.20).

Although it is hard to find q_0 exactly, it is possible to show that there exists finite q_0 such that result of Theorem 3.3 holds for. This can be done by solving above equations assuming that $\epsilon_q(k)$ is zero for every k (assuming $q \gg 1$). Then, it can be observed that the RHS of (3.62) *are either greater or less than zero. Now by assuming finite but large enough q and considering the exact form of* (3.62) *we have some small perturbations that cannot change the sign of RHS of* (3.62) *so we are done.*

3.3.5 Proof of Theorem 3.4

Let $\epsilon_q(k)$ denotes the error term in (3.53). We can easily write the exact expression for $\epsilon_q(k)$ which is as follows

$$\begin{aligned}\epsilon_q(k) = &- \sum_{d_y=0}^{r_k} \psi(N, d_y)\begin{bmatrix} k \\ d_y \end{bmatrix} q^{-Nk} \log\left(\sum_{d_x=d_y}^{\min[M,L]} \alpha_{d_x} \frac{\begin{bmatrix} d_x \\ d_y \end{bmatrix}}{\begin{bmatrix} L \\ d_y \end{bmatrix}} q^{-Nd_x}\right) \\ &+ \log\left(\sum_{d_x=r_k}^{\min[M,L]} q^{r_k(d_x-r_k)-Nd_x}\alpha_{d_x}\right) - r_k(L-r_k)\log q, \end{aligned} \tag{3.73}$$

where $r_k = \min[N, k]$.

We consider the case where $L > N + \min[M, N]$ so Theorem 3.3 implies that for the optimal input distribution we have $\alpha_{i^*} = 1$ where $i^* = \min[M, N]$ and $q > q_0$. Then we can simplify $\epsilon_q(k)$ more and write

$$\epsilon_q(k) = \sum_{d_y=0}^{r_k} \psi(N, d_y) \begin{bmatrix} k \\ d_y \end{bmatrix} q^{-Nk} \log \left(\frac{\begin{bmatrix} L \\ d_y \end{bmatrix}}{\begin{bmatrix} i^* \\ d_y \end{bmatrix}} \right) - r_k(L - i^*) \log q, \tag{3.74}$$

where we also use Lemma 2.7 in the above simplification.

To find q_0, the minimum value of q that the result of Theorem 3.4 is valid for, we should consider the exact form of (3.62) and check that the RHS of (3.62) is less than or equal to zero for $0 \le l \le (i^* - 1)$. So from (3.62) for every $0 \le l \le (i^* - 1)$ we may write

$$\left[1 - q^{(L-N-i^*)(i^*-l)} 2^{[\epsilon_q(i^*) - \epsilon_q(l)]} \right] \le 0, \tag{3.75}$$

or equivalently

$$\frac{\epsilon_{q_0}(l) - \epsilon_{q_0}(i^*)}{(L - N - i^*)(i^* - l)} \le \log q_0, \quad \forall l : 0 \le l \le (i^* - 1). \tag{3.76}$$

Using a similar argument we should have also

$$\frac{\epsilon_{q_0}(l) - \epsilon_{q_0}(i^*)}{i^*(l - i^*)} \le \log q_0, \quad \forall l : (i^* + 1) \le l \le M. \tag{3.77}$$

From (3.71) for the capacity C_s we have $C_s = i^*(L - i^*) \log q + \epsilon_q(i^*)$. Evaluating (3.74) at $k = i^*$ we have

$$\epsilon_q(i^*) = \sum_{d_y=0}^{i^*} \psi(N, d_y) \begin{bmatrix} i^* \\ d_y \end{bmatrix} q^{-Ni^*} \log \left(\frac{\begin{bmatrix} L \\ d_y \end{bmatrix}}{\begin{bmatrix} i^* \\ d_y \end{bmatrix}} \right) - i^*(L - i^*) \log q, \tag{3.78}$$

which results in the capacity stated in the assertion of Theorem 3.4.

Remark 3.3. *We derive a sufficient condition on the minimum size of q to satisfy the set of conditions stated in* (3.76) *and* (3.77)*. Using this sufficient condition we explore the behavior of q_0 as $L \to \infty$.*

For $k \ne i^*$ we can write

$$\begin{aligned} \epsilon_q(k) &\overset{(a)}{\le} 4 \sum_{d_y=0}^{r_k} q^{-(N-d_y)(k-d_y)} \log \left(4q^{d_y(L-i^*)} \right) - r_k(L - i^*) \log q \\ &\le 8 + 4r_k q^{-(\max[N,k] - \min[N,k] + 1)} \left(2 + (r_k - 1)(L - i^*) \log q \right) \\ &\overset{(b)}{\le} 8(1 + r_k) + \left(4r_k(r_k - 1)(L - i^*) \frac{\log q}{q^{(\max[N,k] - \min[N,k] + 1)}} \right), \end{aligned} \tag{3.79}$$

where (a) follows from (2.24) *and* (2.29)*, and in (b) we use the fact that $k \ne i^*$.*

On the other hand, for $k = i^*$ we can write

$$\epsilon_q(i^*) \geq \psi(N, i^*) q^{-Ni^*} \log \begin{bmatrix} L \\ i^* \end{bmatrix} - i^*(L - i^*) \log q$$
$$\stackrel{(a)}{\geq} -(i^*)^2(L - i^*) \frac{\log q}{q^{N-i^*+1}}, \tag{3.80}$$

where (a) follows from (2.24) *and* (2.29).

Let us consider two cases. First, we assume that $M \leq N$ so $i^* = M$. To find a sufficient condition for q_0 we have to only consider conditions given in (3.76). *Using* (3.79) *and* (3.80) *and assuming that $L \to \infty$ we should have $\log q_0 \geq 5M^2 q_0^{-N+M-1} \log q_0$, or equivalently $q_0^{N-M+1} \geq 5(i^*)^2$.*

For the second case we have $M > N$ which means $i^* = N$. Here, using a similar argument to the one given above for the first case we can show that conditions (3.76) *give some constant q_0 as $L \to \infty$. However, the conditions* (3.77) *give a sufficient condition for q_0 which grows as $L \to \infty$. Now, using* (3.77), (3.79), *and* (3.80) *and assuming that $L \to \infty$, a sufficient condition for q_0 would be $\log q_0 \geq 4NLq_0^{-2} \log q_0 + NLq_0^{-1} \log q_0$. For large L for the sufficient condition we have $q_0 \geq i^* L$.*

3.4 Multiple Sources Scenario: The Rate Region

The goal of this section is to characterize $\mathcal{R}$, the set of all achievable rate pairs $(\mathfrak{R}_1, \mathfrak{R}_2)$ for two user communication over the multiple access channel $\mathcal{C}_{m-\mathsf{MAC}}$ described in Definition 3.3. More precisely, we will show that $\mathcal{R} = \mathcal{R}^*$. In order to do this, we first formulate a mathematical model for this channel. Then, we present an achievability scheme, to show that $\mathcal{R}^*$ is achievable, i.e., $\mathcal{R}^* \subseteq \mathcal{R}$. In the next subsection we prove the optimality of this scheme and show that $\mathcal{R} \subseteq \mathcal{R}^*$.

The proof of the converse part of the theorem is based on two outer bounds, namely, a cooperative bound and a coloring bound. For the coloring bound, we utilize a combinatorial argument to bound the number of *distinguishable* symbol pairs that can be transmitted from the two sources to the destination. This bound allows us to restrict the *effective* input alphabets of the sources to subsets of the original alphabets, with significantly smaller size. We can then easily bound the capacity region of the network using the restricted input alphabet.

The transition probability of the channel given by Definition 3.3, $P_{\boldsymbol{Y}|\boldsymbol{X}_1\boldsymbol{X}_2}$, can be written as

$$P_{\boldsymbol{Y}|\boldsymbol{X}_1\boldsymbol{X}_2}(\boldsymbol{y}|\boldsymbol{x}_1, \boldsymbol{x}_2) = \begin{cases} q^{-N \dim(\langle \boldsymbol{x}_1 \rangle + \langle \boldsymbol{x}_2 \rangle)} & \langle \boldsymbol{y} \rangle \sqsubseteq \langle \boldsymbol{x}_1 \rangle + \langle \boldsymbol{x}_2 \rangle, \\ 0 & \text{otherwise.} \end{cases} \tag{3.81}$$

Our first result, stated in Theorem 3.6, is that the multiple access matrix channel described in Definition 3.3 is equivalent to the "subspace" channel

$\mathrm{Ch}_{s-\mathsf{MAC}}$ described in Definition 3.4, that has subspaces as inputs and outputs. So to characterize the optimal rate region of $\mathrm{Ch}_{m-\mathsf{MAC}}$, we can focus on finding the optimal rate region of $\mathrm{Ch}_{s-\mathsf{MAC}}$. We will use this equivalence in the rest of this section.

We know from [38] that the rate region of the multiple access channel $\mathrm{Ch}_{s-\mathsf{MAC}}$ is given by the closure of the convex hull of the rate vectors satisfying

$$\mathfrak{R}_{\mathcal{S}} \leq I(\Pi_{X_{\mathcal{S}}}; \Pi_Y | \Pi_{X_{\mathcal{S}^c}}) \quad \forall \mathcal{S} \subseteq [1:s], \tag{3.82}$$

for some product distribution $P_{\Pi_{X_1}}(\pi_1) \cdots P_{\Pi_{X_s}}(\pi_s)$. Note that $\mathfrak{R}_{\mathcal{S}} = \sum_{i \in \mathcal{S}} \mathfrak{R}_i$, where $\mathfrak{R}_i$ is the transmission rate of the ith source, $\Pi_{X_{\mathcal{S}}} = \{\Pi_{X_i} : i \in \mathcal{S}\}$ and $\mathcal{S}^c$ is the complement set of $\mathcal{S}$.

3.4.1 Achievability Scheme

In this subsection we illustrate a simple achievability scheme for the corner points of the rate region defined in Theorem 3.7. The remaining points in the rate region can be achieved using time-sharing.

For given $(d_1, d_2) \in \mathcal{D}^*$, define the following subspace code-books

$$\widetilde{\mathcal{C}}_1 \triangleq \left\{ \langle \boldsymbol{X}_1 \rangle : \boldsymbol{X}_1 = \left[\begin{array}{c|c|c} \mathbf{I}_{d_1 \times d_1} & \mathbf{0}_{d_1 \times d_2} & \mathbf{U}_1 \\ \hline \mathbf{0}_{(M_1-d_1)\times d_1} & \mathbf{0}_{(M_1-d_1)\times d_2} & \mathbf{0}_{(M_1-d_1)\times(L-d_1-d_2)} \end{array} \right] \right\} \tag{3.83}$$

where $\mathbf{U}_1 \in \mathbb{F}_q^{d_1 \times (L-d_1-d_2)}$, and

$$\widetilde{\mathcal{C}}_2 \triangleq \left\{ \langle \boldsymbol{X}_2 \rangle : \boldsymbol{X}_2 = \left[\begin{array}{c|c|c} \mathbf{0}_{d_2 \times d_1} & \mathbf{I}_{d_2 \times d_2} & \mathbf{U}_2 \\ \hline \mathbf{0}_{(M_2-d_2)\times d_1} & \mathbf{0}_{(M_2-d_2)\times d_2} & \mathbf{0}_{(M_2-d_2)\times(L-d_1-d_2)} \end{array} \right] \right\} \tag{3.84}$$

where $\mathbf{U}_2 \in \mathbb{F}_q^{d_2 \times (L-d_1-d_2)}$.

If the sources transmit messages from these code-books, we have

$$\begin{aligned} \boldsymbol{Y} &= \boldsymbol{H}_1 \boldsymbol{X}_1 + \boldsymbol{H}_2 \boldsymbol{X}_2 \\ &= \left[\begin{array}{c|c|c} \hat{\boldsymbol{H}}_1 & \hat{\boldsymbol{H}}_2 & \hat{\boldsymbol{H}}_1 \mathbf{U}_1 + \hat{\boldsymbol{H}}_2 \mathbf{U}_2 \end{array} \right], \end{aligned} \tag{3.85}$$

where $\hat{\boldsymbol{H}}_i$ captures the first d_i columns of $\boldsymbol{H}_i$. Therefore, decoding at the receiver would be just recovering of $\mathbf{U}_1$ and $\mathbf{U}_2$ given $\hat{\boldsymbol{H}}_1 \mathbf{U}_1 + \hat{\boldsymbol{H}}_2 \mathbf{U}_2$, $\hat{\boldsymbol{H}}_1$, and $\hat{\boldsymbol{H}}_2$. Since $d_1 + d_2 \leq N$ (see 3.30), the matrix $[\hat{\boldsymbol{H}}_1 \ \hat{\boldsymbol{H}}_2]$ is full-rank with high probability, and therefore the decoder is able to decode $\mathbf{U}_1$ and $\mathbf{U}_2$.

Note that the achievability scheme uses effectively the coding vectors approach [20]. This indicates that for $\frac{L}{2} > \max[M_1 + M_2, N]$ and large enough q, the subspace coding and the coding vectors approach achieve the same rate.

3.4.2 Outer bound on the Admissible Rate Region

In the following we will present an outer bound for $\mathcal{R}$, the admissible rate region of the non-coherent two-user multiple access channel $\mathrm{Ch}_{m-\mathsf{MAC}}$. Recall

that by Theorem 3.6 we can focus on the subspace channel $\mathrm{Ch}_{s-\mathsf{MAC}}$. We first show in Proposition 3.1 that $\mathcal{R} \subseteq \mathcal{R}_{\mathrm{coop}}$, a cooperative outer-bound. Then Proposition 3.2 demonstrates that $\mathcal{R} \subseteq \mathcal{R}_{\mathrm{col}}$, a coloring outer-bound. Finally we show that $\mathcal{R}_{\mathrm{col}} \cap \mathcal{R}_{\mathrm{coop}} \subseteq \mathcal{R}$, yielding the desired outer-bound $\mathcal{R} \subseteq \mathcal{R}^*$ which matches the achievability of Section 3.4.1.

The first outer bound, called cooperating outer bound, is simply obtained by letting the two transmitters cooperate to transmit their messages to the receiver, i.e. we assume they form a super-source. Applying Theorem 3.2 for the non-coherent scenario for the single super-source, the one who controls the packets of both transmitters, we have the following proposition.

Proposition 3.1. *Let $L \geq 2(M_1 + M_2)$. Then we have $\mathcal{R} \subseteq \mathcal{R}_{\mathrm{coop}}$ where*

$$\mathcal{R}_{\mathrm{coop}} \triangleq \{(\mathfrak{R}_1, \mathfrak{R}_2) : \ \mathfrak{R}_1 + \mathfrak{R}_2 \leq k(L-k)\log q\} \tag{3.86}$$

and $k = \min[M_1 + M_2, N]$.

The rest of this section is dedicated to deriving the second outer bound which is denoted by $\mathcal{R}_{\mathrm{col}}$. This bound is based on an argument on the number of messages per channel use that each user can reliably communicate over the multiple access channel.

Let $(\mathfrak{R}_1, \mathfrak{R}_2) \in \mathcal{R}$ be an achievable rate pair for which there exists an encoding and decoding scheme with block length n and small error probability. One can follow the usual converse proof of the multiple access channel from [38] to show that

$$\begin{aligned}
\mathfrak{R}_1 &\leq I(\Pi_{X_1}^n; \Pi_Y^n | \Pi_{X_2}^n) \leq \frac{1}{n}\sum_{t=1}^{n} I(\Pi_{X_1 t}; \Pi_{Y t} | \Pi_{X_2 t}), \\
\mathfrak{R}_2 &\leq I(\Pi_{X_2}^n; \Pi_Y^n | \Pi_{X_1}^n) \leq \frac{1}{n}\sum_{t=1}^{n} I(\Pi_{X_2 t}; \Pi_{Y t} | \Pi_{X_1 t}), \\
\mathfrak{R}_1 + \mathfrak{R}_2 &\leq I(\Pi_{X_1}^n, \Pi_{X_2}^n; \Pi_Y^n) \leq \frac{1}{n}\sum_{t=1}^{n} I(\Pi_{X_1 t}, \Pi_{X_2 t}; \Pi_{Y t}).
\end{aligned} \tag{3.87}$$

For each time instance t, denote by $\widetilde{\mathcal{C}}_{i,t}$, the projection of the code-book used by user i to its t-th element. For a single source scenario, we have shown in Section 3.3 that we can use the set $\mathrm{Sp}(L, M)$ as our input alphabet for all time slots, and have the receiver successfully decode the sent messages, and hence, the user can communicate $\mathcal{S}(L, M)$ (see Definition 2.3) distinct messages. For the multi-source case, $\widetilde{\mathcal{C}}_{i,t}$ is more restricted. The main reason for this is that the transition probability of the multiple access channel $P_{\Pi_Y|\Pi_{X_1},\Pi_{X_2}}$ is of the form $P_{\Pi_Y|\Pi_{X_1}+\Pi_{X_2}}$. That is, if $(\pi_1, \pi_2) \in \widetilde{\mathcal{X}}_1 \times \widetilde{\mathcal{X}}_2$ and $(\pi_1', \pi_2') \in \widetilde{\mathcal{X}}_1 \times \widetilde{\mathcal{X}}_2$ satisfy $\pi_1 + \pi_2 = \pi_1' + \pi_2'$, then $P(\Pi_Y|\pi_1, \pi_2) = P(\Pi_Y|\pi_1', \pi_2')$, and hence the receiver cannot distinguish between the two pairs.

In the following we will discuss this indistinguishability in detail, and derive the maximum number of distinguishable pairs which can be conveyed through the channel. In order to do so, we start with some useful definitions and lemmas.

Definition 3.5. *For a fixed $\pi_1 \in \mathrm{Gr}(L, d_1)$, we denote by $\mathcal{N}(\pi_1, d_2, d_{12})$ the set of subspaces of dimension d_2 that intersect with π_1 at d_{12} dimensions,* i.e.*,*

$$\mathcal{N}(\pi_1, d_2, d_{12}) \triangleq \{\pi_2 \in \mathrm{Gr}(L, d_2) : \dim(\pi_1 \cap \pi_2) = d_{12}\}. \tag{3.88}$$

It turns out that the cardinality of the set $\mathcal{N}(\pi_1, d_2, d_{12})$ depends on π_1 only through its dimension, $d_1 = \dim(\pi_1)$. Therefore, we denote this number by $n(d_1, d_2, d_{12})$, which is characterized in the following lemma.

Lemma 3.8. *The cardinality of the set $N(\pi_1, d_2, d_{12})$ is given by*

$$n(d_1, d_2, d_{12}) = |N(\pi_1, d_2, d_{12})| \doteq q^{d_{12}(d_1 - d_{12}) + (d_2 - d_{12})(L - d_2)}. \tag{3.89}$$

Definition 3.6. *For a fixed $\pi_1 \in \mathrm{Gr}(L, d_1)$ and $\pi_2 \in \mathrm{Gr}(L, d_2)$, we define*

$$A(\pi_1, \pi_2) \triangleq \{\pi_2' \in \mathrm{Gr}(L, d_2) : \pi_1 + \pi_2' = \pi_1 + \pi_2\}. \tag{3.90}$$

Lemma 3.9. *The cardinality of the set $A(\pi_1, \pi_2)$ only depends on the dimensions of the two subspaces and their intersection, $d_1 = \dim(\pi_1)$, $d_2 = \dim(\pi_2)$, and $d_{12} = \dim(\pi_1 \cap \pi_2)$. Moreover, it can be asymptotically characterized by*

$$a(d_1, d_2, d_{12}) = |A(\pi_1, \pi_2)| \doteq q^{d_2(d_1 - d_{12})}. \tag{3.91}$$

Definition 3.7. *For an arbitrary set $\widetilde{\mathcal{C}} \subseteq \mathrm{Sp}(L, M)$, we denote the projection of $\widetilde{\mathcal{C}}$ onto the set of d-dimensional Grassmannian $\widetilde{\mathcal{C}}(d)$. Formally,*

$$\widetilde{\mathcal{C}}(d) \triangleq \widetilde{\mathcal{C}} \cap \mathrm{Gr}(L, d) = \{\pi \in \widetilde{\mathcal{C}} : \dim(\pi) = d\}. \tag{3.92}$$

For a fixed time instance t, and corresponding subsets $\widetilde{\mathcal{C}}_{1,t}$ and $\widetilde{\mathcal{C}}_{2,t}$, we can construct a table with $|\widetilde{\mathcal{C}}_{1,t}|$ rows and $|\widetilde{\mathcal{C}}_{2,t}|$ columns, each row (column) corresponding to one subspace π_1 (π_2) in $\widetilde{\mathcal{C}}_{1,t}$ ($\widetilde{\mathcal{C}}_{2,t}$). In the following, we define an equivalence relation for the cells of this table.

Definition 3.8. *A* coloring *for a table constructed as above is an assignment of colors to the cells of the table using a function $\mathrm{col} : \widetilde{\mathcal{C}}_{1,t} \times \widetilde{\mathcal{C}}_{2,t} \to \mathbb{N}$ such that $\mathrm{col}(\pi_1, \pi_2) = \mathrm{col}(\pi_1', \pi_2')$ if and only if $\pi_1 + \pi_2 = \pi_1' + \pi_2'$.*

It is clear that the coloring definition above exactly matches with that of indistinguishability we discussed before. More precisely, two pairs of subspaces (π_1, π_2) and (π_1', π_2') are distinguishable if and only if their corresponding cells in the table have different colors. The following theorem upper bounds the cardinality of the subspace sets based on this fact.

Theorem 3.8. *For each pair of uniquely distinguishable sets $(\widetilde{\mathcal{C}}_{1,t}, \widetilde{\mathcal{C}}_{2,t})$ defined on the input alphabet $\widetilde{\mathcal{X}}_1 \times \widetilde{\mathcal{X}}_2$ for the multiple access channel $\mathrm{Ch}_{s-\mathsf{MAC}}$, there exist integer numbers $0 \leq \delta_i(t) \leq M_i$ such that*

$$|\widetilde{\mathcal{C}}_{i,t}| \dot{\leq} q^{\delta_i(t)(L - \delta_1(t) - \delta_2(t))}, \qquad i = 1, 2. \tag{3.93}$$

Proof. We may drop the time index t in this proof for brevity. For a fixed t, let δ_i be the *dominating* dimension in the set $\widetilde{\mathcal{C}}_i$, i.e.,

$$\delta_i \triangleq \arg\max_d |\widetilde{\mathcal{C}}_i(d)|, \tag{3.94}$$

where $\widetilde{\mathcal{C}}_i(d)$ is as defined in Definition 3.7. It is clear that

$$|\widetilde{\mathcal{C}}_i| = \sum_d |\widetilde{\mathcal{C}}_i(d)| \leq M_i |\widetilde{\mathcal{C}}_i(\delta_i)| \doteq |\widetilde{\mathcal{C}}_i(\delta_i)|, \tag{3.95}$$

where the last asymptotic equality follows from the fact that M_i is a constant with respect to the underlying field size q. This means that we may lose only a constant factor in the code-book size by removing all subspaces from $\widetilde{\mathcal{C}}_1$ ($\widetilde{\mathcal{C}}_2$), except the ones that have dimension δ_1 (δ_2) . Therefore the loss in the rate values would be negligible as q grows. Consider the table constructed for $\widetilde{\mathcal{C}}_1(\delta_1)$ and $\widetilde{\mathcal{C}}_2(\delta_2)$. Let $\pi_1 \in \widetilde{\mathcal{C}}_1(\delta_1)$ be a δ_1-dimensional subspace, and consider the corresponding row of the table. We further partition the columns of the table with respect to π_1 into $\bigcup_{d_{12}=0}^{\min[\delta_1,\delta_2]} \widetilde{\mathcal{C}}_2(\pi_1, \delta_2, d_{12})$, where

$$\widetilde{\mathcal{C}}_2(\pi_1, \delta_2, d_{12}) \triangleq \{\pi_2 \in \widetilde{\mathcal{C}}_2(\delta_2) : \dim(\pi_1 \cap \pi_2) = d_{12}\}. \tag{3.96}$$

We use $K(\pi_1, \delta_2)$ and $K(\pi_1, \delta_2, d_{12})$ to denote the number of different colors in the row that corresponds to π_1 and its intersection with $\widetilde{\mathcal{C}}_2(\pi_1, \delta_2, d_{12})$, respectively.

Note that $\widetilde{\mathcal{C}}_2(\pi_1, \delta_2, d_{12}) \subseteq \mathcal{N}(\pi_1, \delta_2, d_{12})$, and therefore the number of different colors that appear in this partition of the row, cannot exceed the number of colors that could potentially appear if $\mathcal{N}(\pi_1, \delta_2, d_{12}) \subseteq \widetilde{\mathcal{C}}_2$. Recall that $\mathcal{N}(\pi_1, \delta_2, d_{12})$ has $n(\delta_1, \delta_2, d_{12})$ elements, which are split into subsets of size $a(\delta_1, \delta_2, d_{12})$ of the same color. Therefore, for a large field size, the number of different colors in this partition of the row corresponding to π_1, can be upper bounded as

$$K(\pi_1, \delta_2, d_{12}) \leq \frac{n(\delta_1, \delta_2, d_{12})}{a(\delta_1, \delta_2, d_{12})} \doteq q^{(\delta_2 - d_{12})(L - \delta_1 - \delta_2 + d_{12})}. \tag{3.97}$$

Hence,

$$\begin{aligned} K(\pi_1, \delta_2) &= \sum_{d_{12}=0}^{\min[\delta_1,\delta_2]} K(\pi_1, \delta_2, d_{12}) \\ &\dot{\leq} \sum_{d_{12}=0}^{\min[\delta_1,\delta_2]} q^{(\delta_2 - d_{12})(L - \delta_1 - \delta_2 + d_{12})} \\ &\doteq q^{\max_{0 \leq d_{12} \leq \min[\delta_1,\delta_2]} (\delta_2 - d_{12})(L - \delta_1 - \delta_2 + d_{12})} \\ &= q^{\delta_2(L - \delta_1 - \delta_2)} \end{aligned} \tag{3.98}$$

where the asymptotic inequality and equality hold for large q. Moreover, the last equality is based on the assumption $L \geq 2(M_1 + M_2) \geq 2(\delta_1 + \delta_2)$ and the fact that the exponent is a decreasing function of d_{12} for $0 \leq d_{12} \leq \min[\delta_1, \delta_2]$.

It is worth mentioning that this argument holds for each choice of $\pi_1 \in \widetilde{\mathcal{C}}_1(\delta_1)$. This means if the first user transmits a δ_1-dimensional subspace, the receiver cannot distinguish more that $q^{\delta_2(L-\delta_1-\delta_2)}$ different symbols. The same argument holds for a fixed column $\pi_2 \in \widetilde{\mathcal{C}}_2$ which yields an upper bound to the number of distinguishable messages as $q^{\delta_1(L-\delta_1-\delta_2)}$. □

Theorem 3.8 essentially upper bounds the single letter mutual information $I(\Pi_{X_1 t}; \Pi_{Y t} | \Pi_{X_2 t})$ for any time instance t. The following proposition summarizes this discussion.

Proposition 3.2. *We have* $\mathcal{R} \subseteq \mathcal{R}_{\text{col}}$ *where*

$$\mathcal{R}_{\text{col}} \triangleq \text{convex hull} \bigcup_{(d_1,d_2)\in\mathcal{D}_{\text{col}}} \mathcal{R}(d_1, d_2), \tag{3.99}$$

in which $\mathcal{R}(d_1, d_2)$ *is as defined in* (3.28)*, and*

$$\mathcal{D}_{\text{col}} \triangleq \{(d_1, d_2) : \ 0 \leq d_i \leq M_i\}. \tag{3.100}$$

Proof. Using Theorem 3.8, we can upper bound the number of distinguishable pairs for each time instance. For a fixed t, let $\delta_1(t)$ and $\delta_2(t)$ denote the dominating dimensions. Therefore, we have

$$\begin{aligned}
\mathfrak{R}_1 &\leq \frac{1}{n}\sum_{t=1}^{n} I(\Pi_{X_1 t}; \Pi_{Y t} | \Pi_{X_2 t}), \\
&\doteq \frac{1}{n}\sum_{t=1}^{n} \log q^{[\delta_1(t)(L-\delta_1(t)-\delta_2(t))]} \\
&= \frac{1}{n}\sum_{t=1}^{n} \delta_1(t)(L - \delta_1(t) - \delta_2(t)) \log q,
\end{aligned} \tag{3.101}$$

where $0 \leq \delta_i(t) \leq M_i$ for $t = 1, \ldots, n$, and $i = 1, 2$. Similarly, we have

$$\mathfrak{R}_2 \leq \frac{1}{n}\sum_{t=1}^{n} \delta_2(t)(L - \delta_1(t) - \delta_2(t)) \log q. \tag{3.102}$$

Therefore,

$$\begin{aligned}
&(\mathfrak{R}_1, \mathfrak{R}_2) \leq \\
&\frac{1}{n}\sum_{t=1}^{n} \Big\{ \delta_1(t)(L - \delta_1(t) - \delta_2(t)) \log q, \delta_2(t)(L - \delta_1(t) - \delta_2(t)) \log q \Big\}.
\end{aligned} \tag{3.103}$$

It is clear that the RHS of (3.103) is a convex linear combination of the points

$$\Big\{ \delta_1(t)(L - \delta_1(t) - \delta_2(t)) \log q, \delta_1(t)(L - \delta_1(t) - \delta_2(t)) \log q \Big\}_{t=1}^{n} \tag{3.104}$$

which are in the region $\mathcal{R}(\delta_1(t), \delta_2(t))$. This completes the proof. □

Summarizing Proposition 3.1 and Proposition 3.2, we have $\mathcal{R} \subseteq \mathcal{R}_{\text{coop}} \cap \mathcal{R}_{\text{col}}$. So, it only remains to prove the following theorem in order to show that $\mathcal{R}^*$ is an outer bound for the admissible rate region.

Theorem 3.9. *We have* $\mathcal{R}_{\text{coop}} \cap \mathcal{R}_{\text{col}} \subseteq \mathcal{R}^*$.

Before presenting the proof of the theorem, we give the following two lemmas, which help us to characterize the corner points of the region of our interest.

Lemma 3.10. *The set of corner points of* $\mathcal{R}_{\text{col}}$ *is the set of all rate pairs of the form*

$$(\mathfrak{R}_1, \mathfrak{R}_2) = (\mathfrak{R}_1(d_1, d_2), \mathfrak{R}_2(d_1, d_2)), \tag{3.105}$$

for some $(d_1, d_2) \in \widetilde{\mathcal{D}}$, *where*

$$\widetilde{\mathcal{D}} = \Big\{(0, M_2), (1, M_2), \ldots, (M_1, M_2), (M_1, M_2 - 1), \ldots, (M_1, 1), (M_1, 0)\Big\}. \tag{3.106}$$

Lemma 3.11. *If* $\mathcal{R}_{\text{col}} \nsubseteq \mathcal{R}_{\text{coop}}$, *then any intersecting point of*

$$\mathfrak{R}_1 + \mathfrak{R}_2 = k(L - k) \log q \tag{3.107}$$

with the boundary of $\mathcal{R}_{\text{col}}$ *is a point of the form*

$$(\mathfrak{R}_1(d_1, d_2), \mathfrak{R}_2(d_1, d_2)), \tag{3.108}$$

where

$$(d_1, d_2) \in \widetilde{\mathcal{D}} \cup \Big\{(M_1 - 1, 0), \ldots, (0, 0), (0, 1), \ldots, (0, M_2 - 1)\Big\}. \tag{3.109}$$

That is, the boundaries of $\mathcal{R}_{\text{col}}$ *and* $\mathcal{R}_{\text{coop}}$ *can only intersect on either the corner points of* $\mathcal{R}_{\text{col}}$ *or the* $\mathfrak{R}_1$ - $\mathfrak{R}_2$ *axes.*

Proof of Theorem 3.9. Note that $\mathcal{R}_{\text{coop}} \cap \mathcal{R}_{\text{col}}$ is a convex polytope, formed as intersection of a polytope and the convex hull of a finite number of polytopes. Therefore, it suffices to prove the theorem only for its corner points. Let $(\mathfrak{R}_1, \mathfrak{R}_2) \in \mathcal{R}_{\text{coop}} \cap \mathcal{R}_{\text{col}}$ be a corner point. It is clear that one of the followings occurs.

(i) $(\mathfrak{R}_1, \mathfrak{R}_2)$ is a corner point of $\mathcal{R}_{\text{col}}$ and interior point of $\mathcal{R}_{\text{coop}}$;

(ii) $(\mathfrak{R}_1, \mathfrak{R}_2)$ is an intersecting point of the boundaries of $\mathcal{R}_{\text{col}}$ and $\mathcal{R}_{\text{coop}}$.

In the former case, Lemma 3.10 which characterizes the set of corner points of $\mathcal{R}_{\text{col}}$, implies there exists a pair $(d_1, d_2) \in \widetilde{\mathcal{D}}$ such that

$$(\mathfrak{R}_1, \mathfrak{R}_2) = (\mathfrak{R}_1(d_1, d_2), \mathfrak{R}_2(d_1, d_2)). \tag{3.110}$$

Also $(\mathfrak{R}_1, \mathfrak{R}_2) \in \mathcal{R}_{\text{coop}}$ implies

$$(d_1 + d_2)\big(L - (d_1 + d_2)\big) \log q = \mathfrak{R}_1 + \mathfrak{R}_2 \leq k(L - k) \log q. \tag{3.111}$$

Note that the function $f(x) \triangleq x(L-x)$ is an increasing function of x for $x \in (0, L/2)$. Therefore, $d_1 + d_2 \leq k = \min[M_1 + M_2, N]$, and hence $(d_1, d_2) \in \mathcal{D}^*$, which implies that $(\mathfrak{R}_1, \mathfrak{R}_2) \in \mathcal{R}^*$.

In the latter case, it follows from Lemma 3.11 that $(\mathfrak{R}_1, \mathfrak{R}_2)$ should be either a corner point of $\mathcal{R}_{\text{col}}$ for which the above argument holds, or of the form $(\mathfrak{R}_1, \mathfrak{R}_2) = (\mathfrak{R}_1(d_1, d_2), \mathfrak{R}_2(d_1, d_2))$ with $d_1 d_2 = 0$. Again $(\mathfrak{R}_1, \mathfrak{R}_2) \in \mathcal{R}_{\text{coop}}$, which implies that $d_1 + d_2 \leq k = \min[M_1 + M_2, N]$, and $(\mathfrak{R}_1, \mathfrak{R}_2) \in \mathcal{R}^*$. This completes the proof. □

Corollary 3.1. *The number of corner points of the rate region $\mathcal{R}^*$ excluding the point $(0,0)$ is equal to*

$$\min\left[M_1, (N-M_2)^+\right] + \min\left[M_2, (N-M_1)^+\right] + 2 - \mathbb{1}_{\{N \geq M_1 + M_2\}}. \quad (3.112)$$

Proof. By Lemma 3.10 the set of corner points of region $\mathcal{R}_{\text{col}}$ correspond to the pairs (d_1, d_2) which belong to the set

$$\widetilde{\mathcal{D}} = \Big\{(0, M_2), \ldots, (M_1, M_2), \ldots, (M_1, 0)\Big\}. \quad (3.113)$$

In this case the number of corner points excluding $(\mathfrak{R}_1, \mathfrak{R}_2) = (0,0)$ is $M_1 + M_2 + 1$.

However the final rate region is the intersection of $\mathcal{R}_{\text{col}}$ and $\mathcal{R}_{\text{coop}}$, where the later one includes all the rate pairs with sum smaller than $k(L-k)\log q$, $k = \min[M_1 + M_2, N]$, see Proposition 3.1.

Lemma 3.11 explains how these two regions intersect with each other. In this case, the corner points correspond to the pairs (d_1, d_2) which belong to the set

$$\Big\{(0, M_2), \ldots, (\alpha, M_2), (M_1, \beta), \ldots, (M_1, 0)\Big\} \quad (3.114)$$

where $\alpha = \min[M_1, (N-M_2)^+]$ and $\beta = \min[M_2, (N-M_1)^+]$. So the number of corner points excluding $(0,0)$ is

$$\alpha + \beta + 2 - \mathbb{1}_{\{N \geq M_1 + M_2\}}, \quad (3.115)$$

where $\mathbb{1}_{\{N \geq M_1 + M_2\}}$ takes into account the case where two points (α, M_2) and (M_1, β) overlap with each other. □

3.5 Concluding Remarks

In this chapter, we used a random matrix channel to model the problem of multicasting over a packet network that employs randomized network coding. We calculated the capacity of this channel for the case where the finite field of operation $\mathbb{F}_q$ is large, but showed through simulation results fast convergence for small values of q. We prove that use of subspace coding, proposed for algebraic coding in [1, 41], is optimal for this channel. Moreover, we showed that the capacity achieving distribution for very small packet lengths uses subspaces of all dimensions, while as the packet length increases, the number of required

dimensions in the optimal distribution decreases. In particular, the choice of the subspace dimension used in the seminal work of Koetter and Kschischang [1] is indeed optimal for large enough packet size. We extended our work to the case of multiple access with two sources, where we used a coloring argument to derive an outer bound for the capacity that we believe is interesting in itself. We showed that in all the cases we examined, the throughput benefits subspace coding offers as compared to the use of coding vectors go to zero as the alphabet size q increases, and thus use of coding vectors is (asymptotically) optimal.

3.A Omitted Proofs

Proof of Theorem 3.1. To prove the theorem, we start with $I(\boldsymbol{X};\boldsymbol{Y})$ for the channel Ch_m, stated in (3.34), where the channel transition probability is given in (3.33). We will show that for each input distribution $P_{\boldsymbol{X}}(\boldsymbol{x})$ there exists an input distribution $P_{\Pi_X}(\pi_x)$ for the channel Ch_s such that $I(\boldsymbol{X};\boldsymbol{Y}) = I(\Pi_Y;\Pi_X)$ and vice versa.

We know that $P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x}) = P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x}')$ if $\langle \boldsymbol{x}\rangle = \langle \boldsymbol{x}'\rangle$. So we can write

$$I(\boldsymbol{X};\boldsymbol{Y}) = \sum_{\pi_x\in\widetilde{\mathcal{X}},\ \boldsymbol{y}\in\mathcal{Y}} P_{\Pi_X}(\pi_x) P_{\boldsymbol{Y}|\Pi_X}(\boldsymbol{y}|\pi_x) \log\left(\frac{P_{\boldsymbol{Y}|\Pi_X}(\boldsymbol{y}|\pi_x)}{P_{\boldsymbol{Y}}(\boldsymbol{y})}\right), \tag{3.116}$$

where we choose $P_{\Pi_X}(\pi_x) = \sum_{\boldsymbol{x}\in\mathcal{X}:\langle\boldsymbol{x}\rangle=\pi_x} P_{\boldsymbol{X}}(\boldsymbol{x})$ and define

$$P_{\boldsymbol{Y}|\Pi_X}(\boldsymbol{y}|\pi_x) \triangleq \begin{cases} q^{-N\dim(\pi_x)} & \langle\boldsymbol{y}\rangle \sqsubseteq \pi_x, \\ 0 & \text{otherwise.} \end{cases} \tag{3.117}$$

Then expanding $I(\boldsymbol{X};\boldsymbol{Y})$ we have

$$I(\boldsymbol{X};\boldsymbol{Y}) = \sum_{\pi_x\in\widetilde{\mathcal{X}}} P_{\Pi_X}(\pi_x) \sum_{\pi_y\in\widetilde{\mathcal{Y}}} \sum_{\substack{\boldsymbol{y}\in\mathcal{Y},\\ \langle\boldsymbol{y}\rangle=\pi_y}} P_{\boldsymbol{Y}|\Pi_X}(\boldsymbol{y}|\pi_x) \log\left(\frac{P_{\boldsymbol{Y}|\Pi_X}(\boldsymbol{y}|\pi_x)}{P_{\boldsymbol{Y}}(\boldsymbol{y})}\right). \tag{3.118}$$

Now using the symmetry properties of $P_{\boldsymbol{Y}|\Pi_X}(\boldsymbol{y}|\pi_x)$ we can simplify $I(\boldsymbol{X};\boldsymbol{Y})$. In fact $P_{\boldsymbol{Y}|\Pi_X}(\boldsymbol{y}_1|\pi_x) = P_{\boldsymbol{Y}|\Pi_X}(\boldsymbol{y}_2|\pi_x)$ and $P_{\boldsymbol{Y}}(\boldsymbol{y}_1) = P_{\boldsymbol{Y}}(\boldsymbol{y}_2)$ if $\langle\boldsymbol{y}_1\rangle = \langle\boldsymbol{y}_2\rangle$. So we can remove the summation over $\boldsymbol{y}$ and write

$$I(\boldsymbol{X};\boldsymbol{Y}) = \sum_{\pi_x\in\widetilde{\mathcal{X}}} P_{\Pi_X}(\pi_x) \sum_{\pi_y\in\widetilde{\mathcal{Y}}} \psi(L,N,\pi_y) P_{\boldsymbol{Y}|\Pi_X}(\boldsymbol{y}|\pi_x) \log\left(\frac{P_{\boldsymbol{Y}|\Pi_X}(\boldsymbol{y}|\pi_x)}{P_{\boldsymbol{Y}}(\boldsymbol{y})}\right), \tag{3.119}$$

for some matrix $\boldsymbol{y}$ such that $\langle\boldsymbol{y}\rangle = \pi_y$. Recall that $\psi(L,N,\pi_y)$ is defined in Definition 2.4, Section 3.1. Defining

$$P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x) \triangleq \psi(L,N,\pi_y)\ P_{\boldsymbol{Y}|\Pi_X}(\boldsymbol{y}|\pi_x)\big|_{\text{for some } \boldsymbol{y}:\langle\boldsymbol{y}\rangle=\pi_y}, \tag{3.120}$$

we can write

$$\begin{aligned} I(\boldsymbol{X};\boldsymbol{Y}) &= \sum_{\pi_x\in\widetilde{\mathcal{X}},\pi_y\in\widetilde{\mathcal{Y}}} P_{\Pi_X}(\pi_x) P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x) \log\frac{P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x)}{P_{\Pi_Y}(\pi_y)} \\ &= I(\Pi_X;\Pi_Y). \end{aligned} \tag{3.121}$$

Based on the above discussion going back from the channel Ch_s to Ch_m is very easy. It is sufficient to choose

$$P_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{P_{\Pi_X}(\pi_x)}{\psi(L, M, \pi_x)}, \quad \forall \boldsymbol{x} : \ \langle \boldsymbol{x} \rangle = \pi_x, \tag{3.122}$$

for all $\pi_x \in \widetilde{\mathcal{X}}$. This completes the proof. □

Proof of Lemma 2.5. We want to count the number of different matrices $\mathbf{X} \in \mathbb{F}_q^{k \times L}$ such that $\langle \mathbf{X} \rangle = \pi_d$ where π_d is an specific d dimensional subspace of $\mathbb{F}_q^L$.

We know that we can decompose $\mathbf{X}$ as

$$\mathbf{X} = \mathbf{AB}, \quad \mathbf{A} \in \mathbb{F}_q^{k \times d}, \quad \mathbf{B} \in \mathbb{F}_q^{d \times L}, \tag{3.123}$$

where $\mathbf{A}$ and $\mathbf{B}$ are full rank matrices. Let us fix $\mathbf{B}$ such that $\langle \mathbf{B} \rangle = \pi_d$. Now for every two different full rank matrices $\mathbf{A}$ and $\mathbf{A'}$ we would obtain different matrices $\mathbf{X} = \mathbf{AB}$ and $\mathbf{X'} = \mathbf{A'B}$ such that $\mathbf{X} \neq \mathbf{X'}$ and $\langle \mathbf{X} \rangle = \langle \mathbf{X'} \rangle = \pi_d$. So the number of different $\mathbf{X}$ where $\langle \mathbf{X} \rangle = \pi_d$ is equal to the number of full rank $N \times d$ matrices over $\mathbb{F}$ which is equal to $\prod_{i=0}^{d-1}(q^k - q^i)$, and we are done. □

Proof of Lemma 3.4. Let $P_{\Pi_X}(\pi_x)$ be the optimal input distribution of the channel Ch_s with transition probabilities given in (3.6). For a fixed dimension $0 \leq d \leq \min[M, L]$, and an arbitrary permutation

$$\sigma : \left\{1, 2, \ldots, \begin{bmatrix} L \\ d \end{bmatrix}\right\} \longrightarrow \left\{1, 2, \ldots, \begin{bmatrix} L \\ d \end{bmatrix}\right\} \tag{3.124}$$

which acts on subspaces of dimension d, define $P_\sigma(\pi_x)$ as

$$P_\sigma(\pi_x) = \begin{cases} P_{\Pi_X}(\sigma(\pi_x)) & \text{if } \dim(\pi_x) = d, \\ \\ P_{\Pi_X}(\pi_x) & \text{if } \dim(\pi_x) \neq d. \end{cases} \tag{3.125}$$

Also define

$$P^*(\pi_x) = \frac{1}{\begin{bmatrix} L \\ d \end{bmatrix}!} \sum_\sigma P_\sigma(\pi_x) \tag{3.126}$$

where the summation is over all possible permutations. Rewriting the mutual information in (3.43) as a function of the input distribution and the transition probabilities, $I(P_{\Pi_X}(\pi_x), P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x))$, we have

$$\begin{aligned} I(P^*(\pi_x), & P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x)) \\ & = I\left(\frac{1}{\begin{bmatrix} L \\ d \end{bmatrix}!} \sum_\sigma P_\sigma(\pi_x), P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x)\right) \\ & \overset{(a)}{\geq} \frac{1}{\begin{bmatrix} L \\ d \end{bmatrix}!} \sum_\sigma I(P_\sigma(\pi_x), P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x)) \\ & \overset{(b)}{=} I(P_{\Pi_X}(\pi_x), P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x)) \end{aligned} \tag{3.127}$$

where (a) is due to concavity of the mutual information with respect to the input distribution, and (b) holds because

$$I(P_\sigma(\pi_x), P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x)) = I(P_{\Pi_X}(\pi_x), P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x)) \tag{3.128}$$

for all σ, since the permutation only permutes the terms in a summation in (3.43).

Note that $P^*(\pi_x)$ assigns equal probabilities to all subspaces with dimension d, and the above-mentioned inequality shows that it is as good as the optimal input distribution. A similar argument holds for all $0 \leq d \leq \min[M, L]$. Therefore, a dimensional-uniform distribution achieves the capacity of the channel. □

Proof of Lemma 3.5. Assuming an optimal input probability distribution of the form (3.44), the probability of receiving a specific subspace $\Pi_Y = \pi_y$ at the receiver can be written as

$$\begin{aligned} P_{\Pi_Y}(\pi_y) &= \sum_{\pi_x \in \widetilde{\mathcal{X}}} P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x) P_{\Pi_X}(\pi_x) \\ &= \sum_{\pi_x \in \widetilde{\mathcal{X}}:\ \pi_y \sqsubseteq \pi_x} \psi(L, N, \pi_y) q^{-N d_x} \frac{\alpha_{d_x}}{\begin{bmatrix} L \\ d_x \end{bmatrix}}. \end{aligned} \tag{3.129}$$

Splitting the summation into two, we can write

$$P_{\Pi_Y}(\pi_y) = \psi(L, N, \pi_y) \sum_{d_x = d_y}^{\min[M,L]} \sum_{\substack{\pi_x \in \widetilde{\mathcal{X}}: \\ \dim(\pi_x) = d_x, \\ \pi_y \sqsubseteq \pi_x}} \frac{q^{-N d_x} \alpha_{d_x}}{\begin{bmatrix} L \\ d_x \end{bmatrix}}, \tag{3.130}$$

where $d_y = \dim(\pi_y)$. Using Lemma 2.4, we can replace the second summation in (3.130).

Thus we can rewrite (3.130) as follows

$$\begin{aligned} P_{\Pi_Y}(\pi_y) &= \psi(L, N, \pi_y) \sum_{d_x = d_y}^{\min[M,L]} \begin{bmatrix} L - d_y \\ d_x - d_y \end{bmatrix} \frac{q^{-N d_x} \alpha_{d_x}}{\begin{bmatrix} L \\ d_x \end{bmatrix}} \\ &\overset{(a)}{=} \frac{\psi(L, N, \pi_y)}{\begin{bmatrix} L \\ d_y \end{bmatrix}} \sum_{d_x = d_y}^{\min[M,L]} \begin{bmatrix} d_x \\ d_y \end{bmatrix} q^{-N d_x} \alpha_{d_x} \\ &= \frac{\psi(N, d_y)}{\begin{bmatrix} L \\ d_y \end{bmatrix}} \sum_{d_x = d_y}^{\min[M,L]} \begin{bmatrix} d_x \\ d_y \end{bmatrix} q^{-N d_x} \alpha_{d_x}, \end{aligned} \tag{3.131}$$

where (a) follows from Lemma 2.2, where we have

$$\begin{bmatrix} L - d_y \\ d_x - d_y \end{bmatrix} \begin{bmatrix} L \\ d_y \end{bmatrix} = \begin{bmatrix} L \\ d_x \end{bmatrix} \begin{bmatrix} d_x \\ d_y \end{bmatrix}. \tag{3.132}$$

Now we can simplify the mutual information $I(\Pi_X;\Pi_Y)$ in (3.43) as follows. Using (3.6), (3.44), and (3.131) for $I(\Pi_X;\Pi_Y)$ we can write

$$\begin{aligned} I(\Pi_X;\Pi_Y) &= \sum_{\pi_x\in\widetilde{\mathcal{X}},\pi_y\in\widetilde{\mathcal{Y}}} P_{\Pi_X}(\pi_x)P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x)\log\left(\frac{P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x)}{P_{\Pi_Y}(\pi_y)}\right) \\ &= \sum_{d_x=0}^{\min[M,L]}\sum_{d_y=0}^{\min[N,d_x]} \sum_{\substack{\pi_x\in\widetilde{\mathcal{X}}:\\ \dim(\pi_x)=d_x}} \sum_{\substack{\pi_y\in\widetilde{\mathcal{Y}}:\\ \dim(\pi_y)=d_y,\\ \pi_y\sqsubseteq\pi_x}} F(d_x,d_y), \end{aligned} \tag{3.133}$$

where

$$F(d_x,d_y) = \frac{\psi(N,d_y)q^{-Nd_x}\alpha_{d_x}}{\begin{bmatrix}L\\ d_x\end{bmatrix}}\log\left(\frac{q^{-Nd_x}}{f(d_y)}\right) \tag{3.134}$$

and

$$f(d_y) \triangleq \frac{P_{\Pi_Y}(\pi_y)}{\psi(N,d_y)} = \frac{1}{\begin{bmatrix}L\\ d_y\end{bmatrix}}\sum_{d_x=d_y}^{\min[M,L]}\begin{bmatrix}d_x\\ d_y\end{bmatrix}q^{-Nd_x}\alpha_{d_x}, \tag{3.135}$$

because $P_{\Pi_Y}(\pi_y)$ only depends on d_y. Now observe that the two inner most summations depend on π_x and π_y only through their dimensions. So we can write

$$I(\Pi_X;\Pi_Y) = \sum_{d_x=0}^{\min[M,L]}\alpha_{d_x}q^{-Nd_x}\sum_{d_y=0}^{\min[N,d_x]}\psi(N,d_y)\begin{bmatrix}d_x\\ d_y\end{bmatrix}\log\left(\frac{q^{-Nd_x}}{f(d_y)}\right). \tag{3.136}$$

Then using Lemma 2.7 in Section 2.3 we can further simplify the mutual information and write

$$\begin{aligned} I(\Pi_X;\Pi_Y) = &-\sum_{d_x=0}^{\min[M,L]}\alpha_{d_x}Nd_x\log q \\ &-\sum_{d_x=0}^{\min[M,L]}\alpha_{d_x}q^{-Nd_x}\sum_{d_y=0}^{\min[N,d_x]}\psi(N,d_y)\begin{bmatrix}d_x\\ d_y\end{bmatrix}\log(f(d_y)), \end{aligned} \tag{3.137}$$

that is the assertion of Lemma 3.5. □

Proof of Lemma 3.6. By taking the partial derivative of the mutual information with respect to α_k, we have that

$$\begin{aligned} I'_k \triangleq &\frac{\partial I(\Pi_X;\Pi_Y)}{\partial\alpha_k} \\ &= -Nk\log q - \sum_{d_y=0}^{\min[N,k]}\psi(N,d_y)\begin{bmatrix}k\\ d_y\end{bmatrix}q^{-Nk}\log\left(f(d_y)\right) \\ &\quad -\sum_{d_x=0}^{\min[M,L]}\alpha_{d_x}\sum_{d_y=0}^{\min[N,d_x,k]}\psi(N,d_y)\begin{bmatrix}d_x\\ d_y\end{bmatrix}q^{-Nd_x}\frac{\begin{bmatrix}k\\ d_y\end{bmatrix}q^{-Nk}\log e}{\begin{bmatrix}L\\ d_y\end{bmatrix}f(d_y)}. \end{aligned} \tag{3.138}$$

The we can write

$$\begin{aligned} I'_k =& - Nk\log q - \sum_{d_y=0}^{\min[N,k]} \psi(N,d_y) \begin{bmatrix} k \\ d_y \end{bmatrix} q^{-Nk} \log\left(f(d_y)\right) \\ & - \sum_{d_y=0}^{\min[N,k]} \frac{\begin{bmatrix} k \\ d_y \end{bmatrix} \psi(N,d_y) q^{-Nk}}{f(d_y)} \underbrace{\sum_{d_x=d_y}^{\min[M,L]} \alpha_{d_x} \frac{\begin{bmatrix} d_x \\ d_y \end{bmatrix}}{\begin{bmatrix} L \\ d_y \end{bmatrix}} q^{-Nd_x}}_{f(d_y)} \log e \\ \overset{(a)}{=}& - Nk\log q - \sum_{d_y=0}^{\min[N,k]} \psi(N,d_y) \begin{bmatrix} k \\ d_y \end{bmatrix} q^{-Nk} \log\left(f(d_y)\right) - \log e, \end{aligned} \tag{3.139}$$

where to derive (a) we use Lemma 2.7 in Section 2.3. □

Proof of Lemma 3.7. For convenience we rewrite (3.52) again

$$\log\left(f(d_y)\right) = -d_y L \log q + O(q^{-1}) + \log\left(\sum_{d_x=d_y}^{\min[M,L]} q^{-(N-d_y)d_x} \alpha_{d_x}\right). \tag{3.140}$$

We prove the assertion in two steps for every k. First, let us assume that the α_i's are such that we have $\log\left(f(\min[N,k])\right) = o(q)$. Then using (3.140) one can conclude that

$$\sum_{d_x=\min[N,k]}^{\min[M,L]} q^{-(N-d_y)d_x} \alpha_{d_x} = 2^{-o(q)}, \tag{3.141}$$

so we should have $\alpha_i = 2^{-o(q)}$ for $\min[N,k] \le i \le \min[M,L]$. We know that $0 \le \alpha_i \le 1$, and $\sum_{i=0}^{\min[M,L]} \alpha_i = 1$, so $\exists j:\ \alpha_j = \Omega(1)$. So we can deduce that

$$\log(f(d_y)) = \begin{cases} o(q) & j < d_y \le \min[N,k], \\ \Theta(\log q) & 0 \le d_y \le j, \end{cases} \tag{3.142}$$

where j, $0 \le j \le \min[N,k]$, is the largest index such that $\alpha_j = \Omega(1)$. So in this case the dominating term in the summation of (3.51) is the one obtained for $d_y = \min[N,k]$ because the order difference between each term inside the summation of (3.51) is at least of order $\Theta(q)$.

Now, for the second case, let us assume that the α_i's are such that we have $\log\left(f(\min[N,k])\right) = \Omega(q)$. We will show that this assumption leads to a contradiction. Using (3.140) we can write

$$\sum_{d_x=\min[N,k]}^{\min[M,L]} q^{-(N-d_y)d_x} \alpha_{d_x} = 2^{-\Omega(q)}, \tag{3.143}$$

so we should have $\alpha_i = 2^{-\Omega(q)}$ for $\min[N,k] \le i \le \min[M,L]$. As before, we find the asymptotic behavior of $\log(f(d_y))$ for different values of d_y but in this case we should make finer regimes for $\log(f(d_y))$. The asymptotic behavior of α_i, $0 \le i \le \min[N,k]$, is either $2^{-\Omega(q)}$ or $2^{-o(q)}$. So we can write

$$\log(f(d_y)) = \begin{cases} \Omega(q) & l < d_y \le \min[N,k], \\ o(q) & j < d_y \le l, \\ \Theta(\log q) & 0 \le d_y \le j, \end{cases} \tag{3.144}$$

where l, $0 \le l \le \min[N,k]$, is the largest index such that $\alpha_i = 2^{-o(q)}$ which means that $\alpha_i = 2^{-\Omega(q)}$ for $l < i \le \min[M,L]$. As before j, $0 \le j \le \min[N,k]$, is the largest index such that $\alpha_j = \Omega(1)$. Now we check the Kuhn-Tucker conditions, (3.47), for I'_k and I'_j. From the above argument we have that $I'_k = \Omega(q)$ and $I'_j = \Theta(\log q)$. We know that $\alpha_j = \Omega(1) > 0$, so we have $I'_j = \Theta(\log q) = \lambda$. On the other hand, we have $I'_k = \Omega(q) \le \lambda$, which is a contradiction implying the second case cannot occur. This completes the proof. □

Proof of Lemma 3.8. There are $\begin{bmatrix} d_1 \\ d_{12} \end{bmatrix} \doteq q^{d_{12}(d_1-d_{12})}$ different choices for the intersection of π_1 and π_2. We have to choose $d_2 - d_{12}$ basis vectors for the rest of the subspace. This can be done in

$$\frac{\left(q^L - q^{d_1}\right)\left(q^L - q^{d_1+1}\right)\dots\left(q^L - q^{d_1+d_2-d_{12}-1}\right)}{\left(q^{d_2} - q^{d_{12}}\right)\left(q^{d_2} - q^{d_{12}+1}\right)\dots\left(q^{d_2} - q^{d_2-1}\right)} \doteq q^{(d_2-d_{12})(L-d_2)} \tag{3.145}$$

ways. So we have $n(d_1,d_2,d_{12}) \doteq q^{d_{12}(d_1-d_{12})+(d_2-d_{12})(L-d_2)}$.

The proof of this lemma appeared in our paper [42]. An alternative proof can also be obtained from [18, Lemma 2], by proper choice of parameters. □

Proof of Lemma 3.9. Define $\pi = \pi_1 + \pi_2$, where

$$\dim(\pi) = \dim(\pi_1) + \dim(\pi_2) - \dim(\pi_1 \cap \pi_2) = d_1 + d_2 - d_{12} \triangleq d. \tag{3.146}$$

The proof of this lemma is similar to that of Lemma 3.8, unless we can only choose the last $d_2 - d_{12}$ basis vectors from π instead of $\mathbb{F}_q^L$. Therefore replacing L in Lemma 3.8 with d, we have

$$a(\pi_1,\pi_2) \doteq q^{d_{12}(d_1-d_{12})+(d_2-d_{12})(d-d_2)} = q^{d_2(d_1-d_{12})}. \tag{3.147}$$

□

Proof of Lemma 3.10. Let $(\mathfrak{R}_1,\mathfrak{R}_2)$ be a corner point of the region $\mathcal{R}_{\text{col}}$. Since $\mathcal{R}_{\text{col}}$ is the convex hull of a set of primitive regions, there should exist a primitive region $\mathcal{R}(d_1,d_2)$ which contains $(\mathfrak{R}_1,\mathfrak{R}_2)$ as a corner point, i.e.,

$$\exists (d_1,d_2) \in \mathcal{D}_{\text{col}} \quad \text{s.t.} \quad (\mathfrak{R}_1,\mathfrak{R}_2) = \big(\mathfrak{R}_1(d_1,d_2), \mathfrak{R}_2(d_1,d_2)\big). \tag{3.148}$$

We will show that any point $(\mathfrak{R}_1(d_1,d_2), \mathfrak{R}_2(d_1,d_2))$ is dominated by the segment connecting $(\mathfrak{R}_1(d_1+1,d_2), \mathfrak{R}_2(d_1+1,d_2))$ and $(\mathfrak{R}_1(d_1,d_2+1), \mathfrak{R}_2(d_1,d_2+$

1)). In order to show that, we have to prove that there exists some $\lambda \in [0,1]$, such that

$$\begin{aligned}\mathfrak{R}_1(d_1,d_2) &< \lambda\mathfrak{R}_1(d_1+1,d_2) + (1-\lambda)\mathfrak{R}_1(d_1,d_2+1),\\ \mathfrak{R}_2(d_1,d_2) &< \lambda\mathfrak{R}_2(d_1+1,d_2) + (1-\lambda)\mathfrak{R}_2(d_1,d_2+1).\end{aligned} \tag{3.149}$$

After a little simplification, (3.149) can be rewritten as

$$\begin{aligned}\lambda[L-d_1-d_2-1] &< d_1,\\ (1-\lambda)[L-d_1-d_2-1] &< d_2,\end{aligned} \tag{3.150}$$

or

$$\frac{d_1}{L-1-d_1-d_2} < \lambda < \frac{L-1-d_1-2d_2}{L-1-d_1-d_2}. \tag{3.151}$$

The last two inequalities can be satisfied for some choice of λ if and only if $d_1+d_2 < (L-1)/2$. Therefore, if we have $d_1 < M_1$, $d_2 < M_2$, and $d_1+d_2 < (L-1)/2$ for some $(d_1,d_2) \in \mathcal{D}_{\text{col}}$, then (d_1+1,d_2) and (d_1,d_2+1) also belong to $\mathcal{D}_{\text{col}}$, and hence, $(R_1(d_1,d_2),R_2(d_1,d_2))$ is an interior point, and cannot be on the boundary of the region. Eliminating such (d_1,d_2) from $\mathcal{D}_{\text{col}}$, we get $\widetilde{\mathcal{D}}$.

It is also easy to show that all of the rate pairs corresponding to $(d_1,d_2) \in \widetilde{\mathcal{D}}$ are on the boundary of $\mathcal{R}_{\text{col}}$. This can be done by comparing the slope of the connecting segment for two consecutive points (according to the order they are appeared in $\widetilde{\mathcal{D}}$). The slopes are

$$\begin{aligned}&\mathcal{S}\Big\{\big(\mathfrak{R}_1(t,M_2),\mathfrak{R}_2(t,M_2)\big);\big(\mathfrak{R}_1(t+1,M_2),\mathfrak{R}_2(t+1,M_2)\big)\Big\}\\ &\qquad = -\frac{M_2}{L-2t-M_2-1}, \qquad \text{for } 0 \le t \le M_1,\\ &\mathcal{S}\Big\{\big(\mathfrak{R}_1(M_1,t),\mathfrak{R}_2(M_1,t)\big);\big(\mathfrak{R}_1(M_1,t-1),\mathfrak{R}_2(M_1,t-1)\big)\Big\}\\ &\qquad = -\frac{L-2t-M_1-1}{M_1}, \qquad \text{for } 1 \le t \le M_2.\end{aligned} \tag{3.152}$$

It is easy to check that all the slopes are negative and they are in a decreasing order. Therefore, no point in the set $\widetilde{\mathcal{D}}$ can be an interior point. □

Proof of Lemma 3.11. Note that $\mathcal{R}_{\text{col}} \nsubseteq \mathcal{R}_{\text{coop}}$ implies $M_1+M_2 > n$. Since $\mathcal{R}_{\text{col}}$ is a convex region, its boundary intersects with the line $\mathfrak{R}_1+\mathfrak{R}_2 = N(L-N)\log q$ in exactly two points (it cannot be only one point, otherwise it would be inside of $\mathcal{R}_{\text{coop}}$). It is easy to verify that the rate points corresponding to $(d_1,d_2) = ((N-M_2)^+, \min[M_2,N])$ and $(d_1,d_2) = (\min[M_1,N],(N-M_1)^+)$ lie on both the boundary of $\mathcal{R}_{\text{col}}$ and the line $\mathfrak{R}_1+\mathfrak{R}_2 = N(L-N)\log q$. Therefore this line cannot intersect with the boundary of $\mathcal{R}_{\text{col}}$ in any other point. □

3.B Extension to Packet Erasure Networks

Let us write the capacity for the erasure case as follows

$$\begin{aligned} C_e &= \max_{P_{\boldsymbol{X}}} I(\boldsymbol{X};\boldsymbol{Y},N) \\ &= \max_{P_{\boldsymbol{X}}} \left[I(\boldsymbol{X};N) + I(\boldsymbol{X};\boldsymbol{Y}|N)\right] \\ &\overset{(a)}{=} \max_{P_{\boldsymbol{X}}} I(\boldsymbol{X};\boldsymbol{Y}|N) \\ &= \max_{P_{\boldsymbol{X}}} \mathbb{E}_N\left[I(\boldsymbol{X};\boldsymbol{Y})\right], \end{aligned} \tag{3.153}$$

where (a) follows from the independence of input distribution $P_{\boldsymbol{X}}$ and the distribution of the number of received packets P_N.

The Upper Bound:

We can write an upper bound for C_e as follows

$$\begin{aligned} C_e &= \max_{P_{\boldsymbol{X}}} \mathbb{E}_N\left[I(\boldsymbol{X};\boldsymbol{Y})\right] \\ &\leq \mathbb{E}_N\left[\max_{P_{\boldsymbol{X}}} I(\boldsymbol{X};\boldsymbol{Y})\right] \\ &= \mathbb{E}_N\left[i^*(L-i^*)\log q\right], \end{aligned} \tag{3.154}$$

where $i^* = \min[M, N, \lfloor \frac{L}{2} \rfloor]$. From here on let us assume that $M \leq \lfloor \frac{L}{2} \rfloor$. We thus have that $i^* = N$ and we can write

$$C_e \leq \mathbb{E}_N\left[N(L-N)\log q\right]. \tag{3.155}$$

Let us define $\mu_1 \triangleq \mathbb{E}[N]$ and $\mu_2 \triangleq \mathbb{E}\left[N^2\right]$ so we can write

$$C_e \leq (\mu_1 L - \mu_2)\log q. \tag{3.156}$$

The Lower Bound:

For the lower bound we can write

$$\begin{aligned} C_e &= \max_{P_{\boldsymbol{X}}} \mathbb{E}_N\left[I(\boldsymbol{X};\boldsymbol{Y})\right] \\ &\geq \mathbb{E}_N\left[I(\boldsymbol{X};\boldsymbol{Y})\right]_{\text{for some } P_{\boldsymbol{X}}} \\ &= \mathbb{E}_N\left[I(\Pi_X;\Pi_Y)\right]_{\text{for some } P_{\Pi_X}}. \end{aligned} \tag{3.157}$$

From (3.45) we know that we can write

$$\begin{aligned} I(\Pi_X;\Pi_Y) = &- \sum_{d_x=0}^{\min[M,L]} \alpha_{d_x} N d_x \log q \\ &- \sum_{d_x=0}^{\min[M,L]} \alpha_{d_x} q^{-N d_x} \sum_{d_y=0}^{\min[N,d_x]} \psi(N,d_y) \begin{bmatrix} d_x \\ d_y \end{bmatrix} \log(f(d_y)), \end{aligned} \tag{3.158}$$

where

$$f(d_y) \triangleq \frac{1}{\begin{bmatrix} L \\ d_y \end{bmatrix}} \sum_{d_x=d_y}^{\min[M,L]} \begin{bmatrix} d_x \\ d_y \end{bmatrix} q^{-Nd_x} \alpha_{d_x}. \tag{3.159}$$

Now assume that $M \leq \lfloor \frac{L}{2} \rfloor$ and choose the input distribution to be $\alpha_k = 1$ for some $0 \leq k \leq M$ and $\alpha_i = 0$ for all $i \neq k$. Then for this input distribution we have

$$\begin{aligned} I(\Pi_X; \Pi_Y) &= -kN \log q - q^{-kN} \sum_{d_y=0}^{\min[N,k]} \psi(N, d_y) \begin{bmatrix} k \\ d_y \end{bmatrix} \log(f(d_y)) \\ &= -kN \log q - q^{-kN} \sum_{d_y=0}^{\min[N,k]} \psi(N, d_y) \begin{bmatrix} k \\ d_y \end{bmatrix} \log(f(d_y)). \end{aligned} \tag{3.160}$$

Then assuming q is large we may approximate the above mutual information as follows

$$I(\Pi_X; \Pi_Y) \approx -kN \log q - \sum_{d_y=0}^{\min[N,k]} q^{-(N-d_y)(k-d_y)} \log(f(d_y)). \tag{3.161}$$

The term $(N - d_y)(k - d_y)$ in the summation is maximized for $d_y = \min[N, k]$ and because we had shown before in Lemma 3.7 that $\log(f(d_y)) = \Theta(\log q)$, we can write

$$\begin{aligned} I(\Pi_X; \Pi_Y) &\approx -kN \log q - \log(f(\min[N, k])) \\ &\approx -kN \log q - \log\left(q^{\min[N,k](k-L)-Nk}\right) \\ &= \min[N, k](L - k) \log q. \end{aligned} \tag{3.162}$$

So by choosing $k = M$ we can write the lower bound for C_e as follows

$$\begin{aligned} C_e &\geq \mathbb{E}_N \left[I(\Pi_X; \Pi_Y) \right]_{\text{for some } P_{\Pi_X}} \\ &\approx \mathbb{E}_N \left[N(L - M) \log q \right] \\ &= \mu_1 (L - M) \log q. \end{aligned} \tag{3.163}$$

"It doesn't matter how beautiful your theory is, it doesn't matter how smart you are. If it doesn't agree with experiment, it's wrong."

- Richard Feynman

Non-coherent NC: An Arbitrarily Varying Channel Approach

4

As we have seen so far, randomized linear NC [4] is an efficient and practical approach to implement network coding [6, 5] in large dynamically changing networks because it does not require a priori the knowledge of the network topology. However, in order to enable the receivers to decode, to each packet a coding vector is appended to learn the transfer matrix induced by the network.

A different approach, other than using coding vectors, is to assume a non-coherent scenario for communication, as proposed in [1], where neither the source(s) nor the receiver(s) have any knowledge of the network topology or the network nodes operations. Non-coherent communication allows creation of end-to-end systems that are completely oblivious to the network state. In [1], the authors proposed communications via choosing subspaces and they introduced a subspace channel called "operator channel" (a channel which has subspaces as input and output symbols). Then, they focused on algebraic subspace code constructions over a Grassmannian for the operator channel.

Following [1], different probabilistic models have been proposed to model the non-coherent randomized linear NC channel, where these models enable one to define and characterize the capacity for such a channel. In all of these works, when there are no errors in the network, the non-coherent linear NC channel is modeled by a multiplicative matrix channel.

Montanari et al. [25] introduced a probabilistic model to capture the end-to-end functionality of non-coherent NC operation, with a focus on the case of error correction capabilities. Jafari et al. [24, 27, 33] (see also Chapter 3) modeled the non-coherent NC channel by assuming that the transfer matrix has i.i.d. entries selected uniformly at random in every time-slot. They showed that coding over subspaces is sufficient to achieve the capacity. Moreover, they obtained the channel capacity as a solution of a convex optimization problem over $O(\min[M, N])$ variables and when the field size is greater than a threshold,

they characterized the capacity by solving the optimization problem. Silva et al. [26] derived the capacity of the multiplicative finite field matrix channel under the assumption that the transfer matrix is square and chosen uniformly at random among all full-rank matrices. Similarly, in this model the coding over subspaces is sufficient to achieve the capacity. Yang et al. [28, 29] (see also [30, 31]) considered a completely general scenario, making no assumption on the distribution of the transfer matrix. They obtained upper and lower bounds on the channel capacity, and give a sufficient condition on the distribution of the transfer matrix such that coding over subspaces is capacity achieving. They also studied the achievable rates of coding over subspaces. Nobrega et al. [32] considered the case where the probability distribution of the rank of the transfer matrix is arbitrary; however all matrices with the same rank are equiprobable. Then, following an approach similar to Chapter 3 (see also [33]), they expressed the capacity as the solution of a convex optimization problem over $O(\min[M, N])$ variables. They also observed that in this case the subspace codes are sufficient to achieve the capacity.

In most of the previous works (including Chapter 3), only certain probability models for the channel transfer matrix have been discussed. However, in practice a complete probabilistic characterization of the matrix channel is difficult and the network may not follow a given probability model. Instead of assuming a complete probability model, we consider in this chapter that only a partial knowledge about the probabilistic model of the channel is known.

More precisely, we assume that the rank distribution of the transfer matrix is known a priori, but the distribution of matrices among each rank is unknown and arbitrary. Though very similar to the arbitrarily varying channel (AVC) model introduced in [43] (refer to [44] and the references therein), but this non-coherent NC model is not exactly an AVC. We introduce a "partially arbitrary varying channel" (PAVC) to capture the statistical property of this non-coherent NC model.

By extending results for the AVC, we obtain the capacities of the PAVC for randomized and deterministic codes (Theorem 4.1 and 4.3). We further show that the randomized and the deterministic code capacities of the non-coherent NC model are the same (Theorem 4.4), and that subspace coding is sufficient to achieve the capacity (Corollary 4.3). This AVC approach to the non-coherent NC provides a justification for the optimality of subspace coding in a more general setting.

4.1 Problem Setup

In this section, we introduce a non-coherent NC channel model which is different from the model introduced in Chapter 3. Moreover, we introduce the notion of partially arbitrarily varying channel which forms the foundation of the results of this chapter.

4.1.1 Non-coherent Network Coding Channel Model

Consider a unicast communication over a network where the relay nodes perform random linear NC over a finite field $\mathbb{F}_q$. Suppose that time is slotted and the channel is block time-varying. At every time-slot, the source injects M packets $\mathbf{x}_1[t], \ldots, \mathbf{x}_M[t]$ of length L symbols from $\mathbb{F}_q$ into the network, i.e., $\mathbf{X}_i[t] \in \mathbb{F}_q^L$. The receiver collects N packets $\mathbf{y}_1[t], \ldots, \mathbf{y}_N[t]$ and aims to decode the transmitted packets.

We use matrices $\boldsymbol{X}[t]$ and $\boldsymbol{Y}[t]$ to denote respectively, the transmitted and received packets, i.e., the ith rows of these matrices represent the ith transmitted and received packets, respectively. For a unicast communication, at time-slot (block) t, the receiver observes

$$\boldsymbol{Y}[t] = \boldsymbol{H}[t]\boldsymbol{X}[t], \tag{4.1}$$

where $\boldsymbol{X}[t] \in \mathbb{F}_q^{M\times L}$, $\boldsymbol{Y}[t] \in \mathbb{F}_q^{N\times L}$, and $\boldsymbol{H}[t] \in \mathbb{F}_q^{N\times M}$. We assume that the channel transfer matrix $\boldsymbol{H}[t]$ is unknown to both the transmitter and the receiver and it changes arbitrarily from one block to another block with a constraint on its rank. More precisely, the ranks of $\boldsymbol{H}[t]$, $t = 1, 2, \ldots$, are independent and follow the same distribution of a random variable R. The conditional distribution of $\boldsymbol{H}[t]$ given rank $(\boldsymbol{H}[t])$ is unknown and changes arbitrarily for different t. However, we assume that the distribution of the random variable R is known. We may consider the channel transfer matrix as the channel state. For a given sequence of channel transfer matrices $\boldsymbol{h}[1:n]$ the channel transition probability is

$$W_{\mathsf{m}}^n\left(\boldsymbol{y}[1:n]|\boldsymbol{x}[1:n];\boldsymbol{h}[1:n]\right) = \prod_{t=1}^{n} W_{\mathsf{m}}\left(\boldsymbol{y}[t]|\boldsymbol{x}[t];\boldsymbol{h}[t]\right), \tag{4.2}$$

where $W_{\mathsf{m}}(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{h}) \triangleq \mathbb{1}_{\{\boldsymbol{y}=\boldsymbol{x}\boldsymbol{h}\}}$ is a stochastic matrix.

The above model is very similar to an arbitrarily varying channel (AVC) model (refer to [44] for more information about AVC) but it does not completely fit into that model. In this work, we will show that it is indeed possible to extend the AVC concepts and results for the above channel model and characterize its capacity.

4.1.2 Partially Arbitrarily Varying Channel (PAVC)

Before defining a partially arbitrarily varying channel (PAVC), let us first consider an AVC model. Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ denote the input and output symbol of a channel where $\mathcal{X}$ and $\mathcal{Y}$ are finite sets denoting the channel input and output alphabets, respectively. Let us consider a transmission scenario where the channel parameters vary arbitrarily from symbol to symbol during the course of a transmission. More precisely, for the channel transition matrix, we can write

$$W^n(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{s}) \triangleq \prod_{t=1}^{n} W(y_t|x_t;s_t), \tag{4.3}$$

where $\boldsymbol{s} = (s_1, \ldots, s_n)$, $s_i \in \mathcal{S}$, and $W : \mathcal{X} \times \mathcal{S} \to \mathcal{Y}$ is a given stochastic matrix. $\mathcal{S}$ is a finite set, often referred to as the state space. This model, called a "discrete memory-less arbitrarily varying channel," will be referred to as an AVC.

Now, we define a PAVC as an AVC with a probability constraint over the state space $\mathcal{S}$. Define a function $\mathfrak{q} : \mathcal{S} \to \mathcal{Q}$ where $\mathcal{Q} \triangleq \{0, \ldots, m\}$ and define a random variable Q with alphabet $\mathcal{Q}$ whose distribution is known by the encoder and the decoder. For a PAVC, we have $\mathfrak{q}(S_t)$, $t = 1, 2 \ldots$, are independent and follow the same distribution of Q. In other words,

$$P_{\mathfrak{q}(\boldsymbol{S})}(q_1, \ldots, q_n) = \prod_{t=1}^{n} P_Q(q_t), \tag{4.4}$$

where $\mathfrak{q}(\boldsymbol{S}) \triangleq (\mathfrak{q}(S_1), \ldots, \mathfrak{q}(S_n))$. We call this model a "discrete memory-less partially arbitrarily varying channel," and will refer to it as a PAVC.

In this work, we are interested in characterizing the capacity of a PAVC. However, we first have to define the capacity. As there are different notions of capacity for an AVC based on different error criteria, the same is true for a PAVC (for more information refer to [44]).

Suppose that the message set of a code is identified as the set $\mathcal{M} = \{1, \ldots, K\}$, so that a length-$n$ block code is given by a pair of mapping (ψ, ϕ), where $\psi : \mathcal{M} \mapsto \mathcal{X}^n$ is the encoder, and $\phi : \mathcal{Y}^n \mapsto \mathcal{M} \cup \{0\}$ is the decoder, where the output 0 counts for an error. Let us define

$$e(i, \boldsymbol{s}, \psi, \phi) \triangleq \sum_{\boldsymbol{y}:\ \phi(\boldsymbol{y}) \neq i} W^n(\boldsymbol{y}|\psi(i); \boldsymbol{s}). \tag{4.5}$$

Then, the error probability for message i, when this code is used on a PAVC and when the state sequence is given to be $\boldsymbol{s} \in \mathcal{S}^n$, equals

$$e_d(i, \boldsymbol{s}) \triangleq e(i, \boldsymbol{s}, \psi, \phi), \tag{4.6}$$

and the average probability of error for a state sequence $\boldsymbol{s}$ is

$$\bar{e}_d(\boldsymbol{s}) \triangleq \frac{1}{K} \sum_{i=1}^{K} e_d(i, \boldsymbol{s}). \tag{4.7}$$

Definition 4.1. *A number $\mathfrak{R} > 0$ is called an achievable rate for the given PAVC (for deterministic code and average error probability criterion) if for every $\epsilon > 0$, $\delta > 0$, and sufficiently large n, there exists length-n block code (ψ, ϕ) with*

$$\frac{1}{n} \log K > \mathfrak{R} - \delta, \tag{4.8}$$

and

$$\max_{P_{\boldsymbol{S}|\,\mathfrak{q}(\boldsymbol{S})}} \mathbb{E}\left[\bar{e}_d(\boldsymbol{S})\right] \triangleq \max_{P_{\boldsymbol{S}|\,\mathfrak{q}(\boldsymbol{S})}} \sum_{\boldsymbol{s}} \bar{e}_d(\boldsymbol{s}) P_{\boldsymbol{S}|\mathfrak{q}(\boldsymbol{S})}\left(\boldsymbol{s}|\mathfrak{q}(\boldsymbol{s})\right) P_{Q^n}\left(\mathfrak{q}(\boldsymbol{s})\right) \leq \epsilon, \tag{4.9}$$

where $P_{Q^n}(\boldsymbol{q}) \triangleq \prod_{t=1}^{n} P_Q(q_t)$. *The maximum achievable rate is called the capacity of the PAVC and is denoted by* $C_{\mathsf{pavc}}^{\mathsf{d},\mathsf{a}}$ *(where superscript "*a*" denotes for the average error probability criterion given by* (4.7) *and "*d*" denotes for the determinist code).*

Remark 4.1. *Note that if there is no probability constraint on the state space in Definition 4.1 (*$P_{\boldsymbol{S}}$ *is unknown instead of* $P_{\boldsymbol{S}|\mathsf{q}(\boldsymbol{S})}$*), then by replacing the maximization over* $P_{\boldsymbol{S}|\mathsf{q}(\boldsymbol{S})}$ *with* $P_{\boldsymbol{S}}$*, we recover the average error criterion for an AVC, namely,* $\max_{P_{\boldsymbol{S}}} \mathbb{E}\left[\bar{e}_d(\boldsymbol{S})\right] \leq \epsilon$ *is equivalent to* $\max_{\boldsymbol{s}} \bar{e}_d(\boldsymbol{s}) \leq \epsilon$.

In contrast to using deterministic codes, there exists another communication technique called *randomized coding* which can provide improvement in performance if a common source of randomness is available between the source and the destination.

Precisely, a randomized code (Ψ, Φ) is a random variable with values in the family of all length-n block codes (ψ, ϕ), defined earlier in this section, with the same message set $\mathcal{M}$. Then, the error probability for message i, when this code is used on a PAVC and when the state sequence is given to be $\boldsymbol{s} \in \mathcal{S}^n$, equals

$$e_r(i, \boldsymbol{s}) \triangleq \mathbb{E}_{\Psi,\Phi}\left[e(i, \boldsymbol{s}, \Psi, \Phi)\right], \tag{4.10}$$

and the average probability of error for a state sequence $\boldsymbol{s}$ is

$$\bar{e}_r(\boldsymbol{s}) \triangleq \frac{1}{K} \sum_{i=1}^{K} e_r(i, \boldsymbol{s}). \tag{4.11}$$

Similar to Definition 4.1, we define the capacity $C_{\mathsf{pavc}}^{\mathsf{r},\mathsf{a}}$ by replacing the function $\bar{e}_d(\boldsymbol{s})$ with $\bar{e}_r(\boldsymbol{s})$. Here, the superscript "r,a" denotes for *randomized codes* and *average error probability.*

Yet there is another communication scheme called *coding with stochastic encoder* which only allows randomization in the transmitter, i.e., there is no shared randomness between the encoder and the decoder. More precisely, a code with stochastic encoder (Ψ, ϕ) is a random variable with values in the family of all length-n block codes (ψ, ϕ) with the same message set $\mathcal{M}$.

The error probability for message i, when this code is used on a PAVC and when the state sequence is given to be $\boldsymbol{s} \in \mathcal{S}^n$, equals

$$e_t(i, \boldsymbol{s}) \triangleq \mathbb{E}_{\Psi}\left[e(i, \boldsymbol{s}, \Psi, \phi)\right], \tag{4.12}$$

and the average probability of error for a state sequence $\boldsymbol{s}$ is

$$\bar{e}_t(\boldsymbol{s}) \triangleq \frac{1}{K} \sum_{i=1}^{K} e_t(i, \boldsymbol{s}). \tag{4.13}$$

Similar to Definition 4.1, we define the capacity $C_{\mathsf{pavc}}^{\mathsf{t},\mathsf{a}}$ by replacing the function $\bar{e}_d(\boldsymbol{s})$ with $\bar{e}_t(\boldsymbol{s})$. Here, the superscript "t,a" denotes for *codes with stochastic encoder* and *average error probability.*

4.2 Main Results

Our main goal is to characterize the capacity of the non-coherent NC channel described in Section 4.1.1. Toward this end, we first determine the capacity of a general PAVC.

4.2.1 Capacity of a PAVC

Before stating the deterministic code capacity of a PAVC, we need the following definition.

Definition 4.2. *A PAVC is called* symmetrizable *if for some channel* $U : \mathcal{X} \times \mathcal{Q} \mapsto \mathcal{S}$*, and for every* x*,* x'*, and* y *we have*

$$\sum_s W(y|x;s)U\left(s|x',\mathfrak{q}(s)\right)P_Q\left(\mathfrak{q}(s)\right) = \sum_s W(y|x';s)U\left(s|x,\mathfrak{q}(s)\right)P_Q\left(\mathfrak{q}(s)\right). \tag{4.14}$$

Let $\mathcal{U}(\mathcal{X}\times\mathcal{Q}\to\mathcal{S})$ *be the set of all such channel. If* $\mathcal{U}(\mathcal{X}\times\mathcal{Q}\to\mathcal{S}) = \emptyset$ *then the PAVC is called* non-symmetrizable*.*

Then, the following theorem characterizes the capacity of a PAVC for *deterministic* codes and average error criterion.

Theorem 4.1. *For the* deterministic *code capacity* $C_{\mathsf{pavc}}^{\mathsf{d,a}}$ *we have* $C_{\mathsf{pavc}}^{\mathsf{d,a}} > 0$ *if and only if the PAVC is non-symmetrizable. If* $C_{\mathsf{pavc}}^{\mathsf{d,a}} > 0$*, then we have*

$$C_{\mathsf{pavc}}^{\mathsf{d,a}} = \max_{P_X}\min_{P_{S|\mathfrak{q}(S)}} I(P_X,\bar{W}_S) = \min_{P_{S|\mathfrak{q}(S)}}\max_{P_X} I(P_X,\bar{W}_S), \tag{4.15}$$

where

$$\begin{aligned}\bar{W}_S(y|x) &\triangleq \mathbb{E}\left[W(y|x;S)\right]\\ &= \sum_s W(y|x;s)P_{S|\mathfrak{q}(S)}\left(s|\mathfrak{q}(s)\right)P_Q\left(\mathfrak{q}(s)\right),\end{aligned} \tag{4.16}$$

and $I(P_X,\bar{W}_S) \triangleq I(X;Y)$ *such that* Y *is connected to* X *through the channel* $\bar{W}_S$*.*

Proof. For the proof refer to Appendix 4.A. □

Theorem 4.2. *For a PAVC, the capacity of codes with stochastic encoder is equal to the deterministic code capacity, i.e.,* $C_{\mathsf{pavc}}^{\mathsf{t,a}} = C_{\mathsf{pavc}}^{\mathsf{d,a}}$*.*

Proof. For the proof refer to Appendix 4.B. □

Remark 4.2. *Theorem 4.2 shows that randomization at the encoder does not improve the deterministic code capacity of a PAVC.*

The following theorem characterizes the capacity of a PAVC for *randomized* code.

Theorem 4.3. *The* randomized *code capacity of a PAVC is given by*

$$C_{\mathsf{pavc}}^{\mathsf{r},\mathsf{a}} = \max_{P_X} \min_{P_{S|\mathfrak{q}(S)}} I(P_X, \bar{W}_S) = \min_{P_{S|\mathfrak{q}(S)}} \max_{P_X} I(P_X, \bar{W}_S), \tag{4.17}$$

where $\bar{W}_S$ *is defined in* (4.16).

Proof. For the proof refer to Appendix 4.C. □

Remark 4.3. *Same as an AVC, the randomized code capacity of a PAVC for the maximum and the average error probability criteria are the same.*

Remark 4.4. *In a more general scenario, when* $\mathfrak{q}(S_t)$, $t = 1, 2, \ldots$ *are not i.i.d. but still for every time* t *the marginal probability* $\mathbb{P}[\mathfrak{q}(S_t) = i] = P_Q(i)$, *the adversary who controls the channel state has more power and hence the capacity in this case is less than or equal to the capacity of i.i.d. case.*

4.2.2 Capacity of Non-coherent Network Coding

According to the definition of the PAVC in Section 4.1.2, the non-coherent NC model defined by (4.1) is a PAVC for which the deterministic and stochastic code capacities are equal, as stated in Theorem 4.1 and Theorem 4.2, and can be characterized as follows.

Corollary 4.1. *The deterministic and stochastic code capacities of the channel* (4.1) *are equal. They are non-zero and given by*

$$C = \max_{P_{\boldsymbol{X}}} \min_{P_{\boldsymbol{H}|\,\mathrm{rank}(\boldsymbol{H})}} I(\boldsymbol{X}; \boldsymbol{Y}) = \min_{P_{\boldsymbol{H}|\,\mathrm{rank}(\boldsymbol{H})}} \max_{P_{\boldsymbol{X}}} I(\boldsymbol{X}; \boldsymbol{Y}), \tag{4.18}$$

if and only if the channel is non-symmetrizable, i.e., if there is no stochastic matrix $U : \mathcal{X} \times [0 : \min[M, N]] \mapsto \mathcal{H}$ *such that we have*

$$\sum_{r=0}^{\min[M,N]} \sum_{\boldsymbol{h}:\ \mathrm{rank}(\boldsymbol{h})=r} W_{\mathsf{m}}(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{h}) U(\boldsymbol{h}|\boldsymbol{x}', r) P_R(r) = \sum_{r=0}^{\min[M,N]} \sum_{\boldsymbol{h}:\ \mathrm{rank}(\boldsymbol{h})=r} W_{\mathsf{m}}(\boldsymbol{y}|\boldsymbol{x}'; \boldsymbol{h}) U(\boldsymbol{h}|\boldsymbol{x}, r) P_R(r), \tag{4.19}$$

for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{F}_q^{M \times L}$ *and* $\boldsymbol{y} \in \mathbb{F}_q^{N \times L}$.

Similarly, using Theorem 4.3, the randomized code capacity of the non-coherent NC defined by (4.1) is stated in the following corollary.

Corollary 4.2. *The randomized code capacity of the channel defined by* (4.1) *is given by* (4.18).

It is hard to show directly that the channel defined by (4.1) is non-symmetrizable. Instead, we prove this indirectly in the next lemma by showing the existence of a (stochastic) coding scheme that gives a non-zero transmission rate over the channel.

Lemma 4.1. *If $\mathbb{E}[R] > 0$, the channel defined by* (4.1) *is non-symmetrizable, and so by Corollary 4.1, its capacity is non-zero and is given by* (4.18)*. If $\mathbb{E}[R] = 0$, then the capacity is zero.*

Proof. The case for $\mathbb{E}[R] = 0$ follows because $\boldsymbol{H}[t]$ is the zero matrix with probability one. To show the non-symmetrizability of the channel defined by (4.1) when $\mathbb{E}[R] > 0$, we construct a stochastic coding scheme that can achieve a strictly positive rate. The idea is to degrade the channel defined by (4.1) to a binary memory-less Z-channel with a known cross-over probability.

For each time slot t, let $\boldsymbol{G}[t]$ be a random matrix over $\mathbb{F}_q^{M\times 1}$ with uniform i.i.d. components. Define a binary-input binary-output channel as follows. Let $B[t]$ be the input of the channel at time t, which takes the value 0 or 1 in $\mathbb{F}_q$. The output of the channel at the time t is $Y[t] = \operatorname{rank}(\boldsymbol{H}[t]\boldsymbol{G}[t]B[t])$. Since the dimension of the matrix $\boldsymbol{H}[t]\boldsymbol{G}[t]B[t]$ is $N \times 1$, $Y[t]$ takes the integer value 0 or 1. Let us check the transition matrix of this channel. If $B[t] = 0$, then $Y[t] = 0$. If $B[t] = 1$, then $Y[t] = \operatorname{rank}(\boldsymbol{H}[t]\boldsymbol{G}[t])$. Note that $\operatorname{rank}(\boldsymbol{H}[t]\boldsymbol{G}[t])$ is a random variable whose distribution only depends on the distribution of $\operatorname{rank}(\boldsymbol{H}[t]) \sim R$ (see the computation in [28, Section IV]). Since $\operatorname{rank}(\boldsymbol{H}[t])$, $t = 1, 2, \ldots$ are independent, the channel is a binary memory-less Z channel.

What remains is to check the cross over probability of the Z channel given by

$$\mathbb{P}[Y[t] = 0 | X[t] = 1] = \mathbb{P}[\operatorname{rank}(\boldsymbol{H}[t]\boldsymbol{G}[t]) = 0]. \tag{4.20}$$

Since $\mathbb{E}[\operatorname{rank}(\boldsymbol{H}[t])] = \mathbb{E}[R] > 0$, $\mathbb{P}[\operatorname{rank}(\boldsymbol{H}[t]\boldsymbol{G}[t]) = 0] < 1$, because otherwise $\boldsymbol{H}[t]$ is the zero matrix with probability one, a contradiction to the assumption that $\mathbb{E}[R] > 0$. Hence, the channel has a positive capacity. □

Definition 4.3 ([32])**.** *A random matrix is called* u.g.r. *(uniform given rank) if any two matrices with the same rank are equiprobable.*

Lemma 4.2. *For any $N \times M$ random matrix $\boldsymbol{H}$, $\boldsymbol{AHB}$ is u.g.r. with the same rank distribution as of $\boldsymbol{H}$, where $\boldsymbol{A} \sim \mathsf{Uni}\left(\mathbb{F}_q^{N\times N}, N\right)$ and $\boldsymbol{B} \sim \mathsf{Uni}\left(\mathbb{F}_q^{M\times M}, M\right)$ are uniform and full-rank random matrices, and $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{H}$ are independent.*

Proof. Let $\boldsymbol{G} = \boldsymbol{AHB}$. Then

$$P_{\boldsymbol{G}}(\boldsymbol{g}) = \sum_{\substack{\boldsymbol{a}\in\mathbb{F}_q^{N\times N}, \boldsymbol{b}\in\mathbb{F}_q^{M\times M},\\ \operatorname{rank}(\boldsymbol{a})=N, \operatorname{rank}(\boldsymbol{b})=M}} P_{\boldsymbol{A}}(\boldsymbol{a}) P_{\boldsymbol{B}}(\boldsymbol{b}) P_{\boldsymbol{H}}(\boldsymbol{a}^{-1}\boldsymbol{g}\boldsymbol{b}^{-1}), \tag{4.21}$$

where $P_{\boldsymbol{A}}(\boldsymbol{a})$ and $P_{\boldsymbol{B}}(\boldsymbol{b})$ respectively do not depend on $\boldsymbol{a}$ and $\boldsymbol{b}$. Now, for another instance $\boldsymbol{g}'$ of $\boldsymbol{G}$ with $\boldsymbol{g}' = \boldsymbol{U}\boldsymbol{g}\boldsymbol{V}$ for some full rank matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, we can see that $P_{\boldsymbol{G}}(\boldsymbol{g}) = P_{\boldsymbol{G}}(\boldsymbol{g}')$. In the following we show that if $\operatorname{rank}(\boldsymbol{g}) = \operatorname{rank}(\boldsymbol{g}')$, then there exist full rank matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ such that $\boldsymbol{g}' = \boldsymbol{U}\boldsymbol{g}\boldsymbol{V}$.

Fix two decompositions $\boldsymbol{g} = \boldsymbol{bc}$ and $\boldsymbol{g}' = \boldsymbol{b}'\boldsymbol{c}'$ with $\operatorname{rank}(\boldsymbol{b}) = \operatorname{rank}(\boldsymbol{b}') = \operatorname{rank}(\boldsymbol{g})$, which implies $\operatorname{rank}(\boldsymbol{c}) = \operatorname{rank}(\boldsymbol{c}') = \operatorname{rank}(\boldsymbol{g})$. Then there exist full rank square matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ such that $\boldsymbol{Ub} = \boldsymbol{b}'$ and $\boldsymbol{cV} = \boldsymbol{c}'$. Hence, $\boldsymbol{g}' = \boldsymbol{UgV}$. □

Lemma 4.3. *In the capacity expression* (4.18)*, the u.g.r. distribution for* $P_{\boldsymbol{H}|\operatorname{rank}(\boldsymbol{H})}$ *is a minimizer for the expression.*

Proof. Let $P^*_{\boldsymbol{H}|\operatorname{rank}(\boldsymbol{H})}$ be the distribution that minimizes (4.18). Now consider a new channel defined by $\boldsymbol{AHB}$ where $\boldsymbol{A} \sim \mathsf{Uni}\left(\mathbb{F}_q^{N\times N}, N\right)$ and $\boldsymbol{B} \sim \mathsf{Uni}\left(\mathbb{F}_q^{M\times M}, M\right)$ are uniform full rank random matrices (note that $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{H}$ are independent). Then by Lemma 4.2, the rank distribution of $\boldsymbol{AHB}$ is the same as that of $\boldsymbol{H}$, but $\boldsymbol{AHB}$ has a u.g.r. distribution.

By the data processing inequality, the mutual information between the input and output of the new channel is less than or equal to the original channel. So if $P^*_{\boldsymbol{H}|\operatorname{rank}(\boldsymbol{H})}$ is a minimizer, then the u.g.r. distribution with the same rank distribution is also a minimizer. □

From Corollary 4.1, Corollary 4.2, Lemma 4.1, and Lemma 4.3 we obtain the following theorem.

Theorem 4.4. *The randomized and deterministic code capacities of the non-coherent NC model, i.e., the matrix channel defined by* (4.1)*, are the same and are equal to the capacity of the matrix channel* $\boldsymbol{Y} = \bar{\boldsymbol{H}}\boldsymbol{X}$ *where* $\bar{\boldsymbol{H}}$ *has the same rank distribution as* $\boldsymbol{H}$ *but has uniform distribution among matrices having the same rank, i.e.,*

$$C = \max_{P_{\boldsymbol{X}}} \min_{P_{\boldsymbol{H}|\operatorname{rank}(\boldsymbol{H})}} I(\boldsymbol{X};\boldsymbol{Y}) = \max_{P_{\boldsymbol{X}}} I(\boldsymbol{X};\bar{\boldsymbol{H}}\boldsymbol{X}).$$

Theorem 4.4 shows that, if only the knowledge of the rank distribution of the transfer matrix is available, the maximum rate that we can communicate over the channel defined by (4.1) is equal to the communication rate over a channel which has the same rank distribution but the channel transfer matrix is u.g.r.

Now, it is shown in [32, Theorem 16] that for a matrix multiplicative channel with u.g.r. distribution over the transfer matrix, the subspace coding is sufficient to achieve the capacity. So we have the following corollary.

Corollary 4.3. *Subspace coding is sufficient to achieve the capacity (randomized and deterministic) of the non-coherent NC channel discussed in Section 4.1.1.*

Although determining the exact value of the capacity in Theorem 4.4 is still open, as shown in [32], the capacity can be expressed as the solution of a convex optimization problem with only $O\left(\min[M,N]\right)$ parameters which is computationally tractable.

4.3 Concluding Remarks

In this chapter, we proposed an arbitrarily varying channel (AVC) approach to model the non-coherent NC by a matrix channel where the channel statistics is known only up to a rank distribution over the transfer matrix.

The previous works investigate the capacity of non-coherent network coding (modeled by the matrix channel) for certain probability distributions. In contrast, we relax the problem model by considering that only the rank distribution of the transfer matrix is known and apart from that the transfer matrix can be changed arbitrarily from time-slot to time-slot. We believe that this AVC approach better fits to model complex networks where relay nodes perform randomized NC.

In order to characterize the capacity of such a channel, we defined a new class of channels, called partially AVC (PAVC), with a partial probabilistic constraint over the state space. By extending the previous result on AVC to PAVC, we proved that the subspace coding is optimal to achieve the capacity of non-coherent NC.

4.A Deterministic Code Capacity of a PAVC: Proof of Theorem 4.1

In this section, we prove Theorem 4.1. The proof goes along similar steps as it goes in [45]. However, for completeness, we will be going to write the whole steps here.

Let us start with some definitions. For $\eta \geq 0$, let us define a family of joint distribution P_{XSY} of random variables X, S, and Y with values from the sets $\mathcal{X}$, $\mathcal{S}$, and $\mathcal{Y}$, respectively, by

$$\mathscr{D}_\eta \triangleq \{P_{XSY} : \ D\left(P_{XSY} || P_X \times P_S \times W\right) \leq \eta\}, \tag{4.22}$$

where $P_S(s) = P_Q(\mathfrak{q}(s)) \times P_{S|\mathfrak{q}(S)}(s|\mathfrak{q}(s))$, $D(\cdot||\cdot)$ denotes Kullback-Leibler information divergence, and $P_X \times P_Q \times P_{S|\mathfrak{q}(S)} \times W$ denotes a joint distribution on $\mathcal{X}\times\mathcal{S}\times\mathcal{Y}$ with probability mass function $P_X(x)P_Q(\mathfrak{q}(s))P_{S|\mathfrak{q}(S)}(s|\mathfrak{q}(s))W(y|x;s)$. Note that in the above definitions, P_Q is known and fix for a particular PAVC. We also define, for any distribution P on $\mathcal{X}$, the quantity

$$I(P) \triangleq \min_{\substack{P_{S|\mathfrak{q}(S)}: \\ P_{XSY}\in\mathscr{D}_0,\ P_X=P}} I(X;Y), \tag{4.23}$$

where $\mathscr{D}_0$ denotes $\mathscr{D}_\eta$ for $\eta = 0$.

From [46], we define the *type* of a sequence $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{X}^n$ to be the distribution $P_{\boldsymbol{x}}$ on $\mathcal{X}$ where $P_{\boldsymbol{x}}(a)$ is the relative frequency of $a \in \mathcal{X}$ in $\boldsymbol{x}$. Similarly, *joint types* are distributions on product spaces. Joint types of length-n sequences will be represented by joint distributions of dummy random variables. For example, if X, S, Y represents a joint type, i.e., $P_{XSY} = P_{\boldsymbol{x},\boldsymbol{s},\boldsymbol{y}}$ for some $\boldsymbol{x} \in \mathcal{X}^n$, $\boldsymbol{s} \in \mathcal{S}^n$, and $\boldsymbol{y} \in \mathcal{Y}^n$, we write

$$\begin{aligned}
\mathsf{T}_X &\triangleq \{\boldsymbol{x} : \ \boldsymbol{x} \in \mathcal{X}^n, P_{\boldsymbol{x}} = P_X\}, \\
\mathsf{T}_{XY} &\triangleq \{(\boldsymbol{x},\boldsymbol{y}) : \ \boldsymbol{x} \in \mathcal{X}^n, \boldsymbol{y} \in \mathcal{Y}^n, P_{\boldsymbol{x},\boldsymbol{y}} = P_{XY}\}, \\
\mathsf{T}_{XSY} &\triangleq \{(\boldsymbol{x},\boldsymbol{s},\boldsymbol{y}) : \ \boldsymbol{x} \in \mathcal{X}^n, \boldsymbol{s} \in \mathcal{S}^n, \boldsymbol{y} \in \mathcal{Y}^n, P_{\boldsymbol{x},\boldsymbol{s},\boldsymbol{y}} = P_{XSY}\}.
\end{aligned} \tag{4.24}$$

Similarly, we use notation for sections of T_{XY}, T_{XSY}, etc.; for example

$$\begin{aligned}
\mathsf{T}_{Y|X}(\boldsymbol{x}) &\triangleq \{\boldsymbol{y} : \ (\boldsymbol{x},\boldsymbol{y}) \in \mathsf{T}_{XY}\}, \\
\mathsf{T}_{Y|XS}(\boldsymbol{x},\boldsymbol{s}) &\triangleq \{\boldsymbol{y} : \ (\boldsymbol{x},\boldsymbol{s},\boldsymbol{y}) \in \mathsf{T}_{XSY}\}.
\end{aligned} \tag{4.25}$$

Lemma 4.4. *If the PAVC is non-symmetrizable (see Definition 4.2), then $I(P)$ defined by (4.23) is positive for every P satisfying $P(x) > 0$ for all $x \in \mathcal{X}$.*

Proof. In fact, if $I(P)$ were zero for such a P, then (4.23) implies the existence of random variable S such that for $P_{XSY} = P_X P_Q P_{S|\mathfrak{q}(S)} W$, X and Y are independent. Thus, we have

$$\sum_{s\in\mathcal{S}} W(y|x;s) P_{S|\mathfrak{q}(S)}(s|\mathfrak{q}(s)) P_Q(\mathfrak{q}(s)) = P_Y(y), \tag{4.26}$$

which does not depend on x. This implies the symmetrizability of the channel in a trivial manner, with $U(s|x,q) = P_{S|\mathfrak{q}(S)}(s|q)$, which leads to a contradiction. □

Now, the proof of Theorem 4.1 proceeds as follows.

Proof of Theorem 4.1. First, note that by [47, Lemma 3.1] we have

$$\max_{P_X} \min_{P_{S|\mathfrak{q}(S)}} I(P_X, \bar{W}_S) = \min_{P_{S|\mathfrak{q}(S)}} \max_{P_X} I(P_X, \bar{W}_S). \tag{4.27}$$

The converse part of this theorem follows by applying Lemma 4.5 and Lemma 4.6.

By Lemma 4.4, non-symmetrizability implies that $I(P) > 0$ for every strictly positive P. In order to prove that for a non-symmetrizable PAVC, $\max_P I(P)$ is an achievable rate, we use the continuity of $I(P)$ as a function of P and by applying Lemma 4.12, we conclude the achievability part of Theorem 4.1. □

The following lemma, Lemma 4.5, is similar to [45, Lemma 1] and describes the converse part of the proof when the channel is symmetrizable.

Lemma 4.5. *For a symmetrizable PAVC, any deterministic code of block length n with $K \geq 2$ codewords, each of type P has*

$$\mathbb{E}\left[\bar{e}_d(\boldsymbol{S})\right] = \max_{P_{\boldsymbol{S}|\mathfrak{q}(\boldsymbol{S})}} \sum_{\boldsymbol{s}\in\mathcal{S}^n} \bar{e}_d(\boldsymbol{s}) P_{\boldsymbol{S}|\mathfrak{q}(\boldsymbol{S})}(\boldsymbol{s}|\mathfrak{q}(\boldsymbol{s})) P_{Q^n}(\mathfrak{q}(\boldsymbol{s})) \geq \frac{1}{4}. \tag{4.28}$$

Proof. Consider an arbitrary code with codeword set $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\}$ and decoder ϕ, where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{in})$ for $i \in [1:K]$. For some $U \in \mathcal{U}(\mathcal{X} \times \mathcal{Q} \to \mathcal{S})$ satisfying (4.14) consider K random sequences $\boldsymbol{S}_j = (S_{j1}, \ldots, S_{jn})$ where $\boldsymbol{S}_j \in \mathcal{S}^n$, with statistically independent components, where

$$\mathbb{P}\left[S_{jk} = s\right] = U(s|x_{jk}, \mathfrak{q}(s)) P_Q(\mathfrak{q}(s)). \tag{4.29}$$

Then for each pair (i,j) and every $\boldsymbol{y} = (y_1, \ldots, y_n) \in \mathcal{Y}^n$ we can write

$$\begin{aligned}\mathbb{E}\left[W^n(\boldsymbol{y}|\boldsymbol{x}_i, \boldsymbol{S}_j)\right] &= \prod_{k=1}^{n} \mathbb{E}\left[W(y_k|x_{ik}, S_{jk})\right] \\ &= \prod_{k=1}^{n} \sum_{s\in\mathcal{S}} W(y_k|x_{ik}, s) U(s|x_{jk}, \mathfrak{q}(s)) P_Q(\mathfrak{q}(s)).\end{aligned} \tag{4.30}$$

So, by using (4.14), it follows that

$$\mathbb{E}\left[W^n(\boldsymbol{y}|\boldsymbol{x}_i, \boldsymbol{S}_j)\right] = \mathbb{E}\left[W^n(\boldsymbol{y}|\boldsymbol{x}_j, \boldsymbol{S}_i)\right], \tag{4.31}$$

and hence for $i \neq j$ we have

$$\begin{aligned}\mathbb{E}\left[e_d(i, \boldsymbol{S}_j)\right] + \mathbb{E}\left[e_d(j, \boldsymbol{S}_i)\right] &= \\ &= \sum_{\boldsymbol{y}:\ \phi(\boldsymbol{y})\neq i} \mathbb{E}\left[W^n(\boldsymbol{y}|\boldsymbol{x}_i; \boldsymbol{S}_j)\right] + \sum_{\boldsymbol{y}:\ \phi(\boldsymbol{y})\neq j} \mathbb{E}\left[W^n(\boldsymbol{y}|\boldsymbol{x}_j; \boldsymbol{S}_i)\right] \\ &\geq \sum_{\boldsymbol{y}\in\mathcal{Y}^n} \mathbb{E}\left[W^n(\boldsymbol{y}|\boldsymbol{x}_i; \boldsymbol{S}_j)\right] \\ &= 1. \end{aligned} \tag{4.32}$$

Now, using this fact we can write

$$\begin{aligned}\frac{1}{K}\sum_{j=1}^{K} \mathbb{E}\left[\bar{e}_d(\boldsymbol{S}_j)\right] = \frac{1}{K^2}\sum_{i=1}^{K}\sum_{j=1}^{K} \mathbb{E}\left[e_d(i, \boldsymbol{S}_j)\right] \\ \geq \frac{1}{K^2}\cdot\frac{K(K-1)}{2} \\ = \frac{K-1}{2K}, \end{aligned} \tag{4.33}$$

so it follows that for some $j \in [1:K]$ we have

$$\mathbb{E}\left[\bar{e}_d(\boldsymbol{S}_j)\right] \geq \frac{K-1}{2K} \geq \frac{1}{4}. \tag{4.34}$$

This leads to the desired result because $\mathbb{E}\left[\bar{e}_d(\boldsymbol{S})\right] \geq 1/4$ for some distribution over $\boldsymbol{S}$ such that the kth element of the random sequence $\boldsymbol{S}$ is distributed independently according to the distribution of the form $P_{S|\mathfrak{q}(S)}P_Q$ where $P_{S|\mathfrak{q}(S)}(s|q) = U(s|x_{jk}, q)$. So in general we have $\max_{P_{\boldsymbol{S}|\mathfrak{q}(\boldsymbol{S})}} \mathbb{E}\left[\bar{e}_d(\boldsymbol{S})\right] \geq 1/4$. □

The following lemma, Lemma 4.6, is similar to [45, Lemma 2] and describes the converse part of the proof when the rate is greater than $I(P)$.

Lemma 4.6. *For any $\delta > 0$ and $\epsilon < 1$, there exists n_0 such that for any code of block length $n \geq n_0$ with codewords, each of type P, $\frac{1}{n}\log K \geq I(P) + \delta$ implies*

$$\mathbb{E}\left[\bar{e}_d(\boldsymbol{S})\right] = \max_{P_{\boldsymbol{S}|\mathfrak{q}(\boldsymbol{S})}} \sum_{\boldsymbol{s}\in\mathcal{S}^n} \bar{e}_d(\boldsymbol{s}) P_{\boldsymbol{S}|\mathfrak{q}(\boldsymbol{S})}(\boldsymbol{s}|\mathfrak{q}(\boldsymbol{s})) P_{Q^n}(\mathfrak{q}(\boldsymbol{s})) > \epsilon. \tag{4.35}$$

Proof. Suppose that $P^*_{S|\mathfrak{q}(S)}$ achieves the minimum in (4.23). So for

$$P_{XSY}(x, s, y) = P(x)P_Q(\mathfrak{q}(s))P^*_{S|\mathfrak{q}(S)}(s|\mathfrak{q}(s))W(y|x; s) \tag{4.36}$$

we have $I(X;Y) = I(P)$.

Now consider any code with codewords $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\}$ and decoder ϕ, and let $\boldsymbol{S} = (S_1, \ldots, S_n)$ be n independent realization of S according to the distribution

$P^*_{S|\mathfrak{q}(S)}P_Q$. Then we can write

$$\begin{aligned}\mathbb{E}\left[\bar{e}_d(\boldsymbol{S})\right] &= \frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[e_d(i,\boldsymbol{S})\right]\\ &= \frac{1}{K}\sum_{i=1}^{K}\sum_{\boldsymbol{y}:\phi(\boldsymbol{y})\neq i}\mathbb{E}\left[W^n(\boldsymbol{y}|\boldsymbol{x}_i;\boldsymbol{S})\right]\\ &= \frac{1}{K}\sum_{i=1}^{K}\sum_{\boldsymbol{y}:\phi(\boldsymbol{y})\neq i}\prod_{j=1}^{n}\mathbb{E}\left[W(y_j|x_{ij};S_j)\right]. \end{aligned} \tag{4.37}$$

If we introduce a new discrete memory-less channel (DMC) $\bar{W}_S$ defined by

$$\bar{W}_S(y|x) = \mathbb{E}\left[W(y|x;S)\right] = \sum_{s\in\mathcal{S}} W(y|x;s)P_{S|\mathfrak{q}(S)}(s|\mathfrak{q}(s))P_Q(\mathfrak{q}(s)), \tag{4.38}$$

then we have $\mathbb{E}\left[\bar{e}_d(\boldsymbol{S})\right] = \bar{e}_{(\bar{W}_S)}$, where $\bar{e}_{(\bar{W}_S)}$ is the average probability of error when the given code is used on the DMC $\bar{W}_S$.

Now, notice that (4.36) means that Y is connected to X by the channel $\bar{W}_S$. As mentioned before, we have $I(X;Y) = I(P)$ so by the strong converse to the coding theorem for a DMC with codewords of type P (see [46, Corollary 1.4, p.104]), $\bar{e}_{(\bar{W}_S)}$ is arbitrary close to 1 if $\frac{1}{n}\log K \geq I(P)+\delta$ and n is large enough. This completes the proof of Lemma 4.6. □

In order to prove the achievability part of Theorem 4.1, we need to define a suitable decoder ϕ. Here, we will use the same decoder as introduced in [45, Definition 3].

Definition 4.4 ([45, Definition 3])**.** *Given the codewords $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_K\}$, let $\phi(\boldsymbol{y}) = i$ if and only if an $\boldsymbol{s}\in\mathcal{S}^n$ exists such that*

1. *the joint type $P_{\boldsymbol{x}_i,\boldsymbol{s},\boldsymbol{y}}$ belongs to $\mathscr{D}_\eta$;*
2. *for each competitor $j\neq i$, such that $P_{\boldsymbol{x}_j,\boldsymbol{s}',\boldsymbol{y}}\in\mathscr{D}_\eta$ for some $\boldsymbol{s}'\in\mathcal{S}^n$, we have $I(XY;X'|S)\leq\eta$, where X,X',S,Y denote dummy random variables such that $P_{XX'SY} = P_{\boldsymbol{x}_i,\boldsymbol{x}_j,\boldsymbol{s},\boldsymbol{y}}$.*

If no such i exists, we set $\phi(\boldsymbol{y}) = 0$, i.e., declare an error.

Before proceeding further, let us state the following lemmas (Lemma 4.7-Lemma 4.9) which are some basic bounds on types (e.g., see [46, Chapter 1]).

Lemma 4.7. *The number of possible joint types of sequences of length n is a polynomial in n.*

Lemma 4.8. *If $\mathsf{T}_X\neq\emptyset$, we have*

$$(n+1)^{-|\mathcal{X}|}\exp\{nH(X)\}\leq|\mathsf{T}_X|\leq\exp\{nH(X)\}, \tag{4.39}$$

and if $\mathsf{T}_{Y|X}(\boldsymbol{x})\neq\emptyset$, we have

$$(n+1)^{-|\mathcal{X}||\mathcal{Y}|}\exp\{nH(Y|X)\}\leq|\mathsf{T}_{Y|X}(\boldsymbol{x})|\leq\exp\{nH(Y|X)\}. \tag{4.40}$$

Lemma 4.9. *For any channel $V : \mathcal{X} \mapsto \mathcal{Y}$, we have*

$$\sum_{\boldsymbol{y} \in \mathsf{T}_{Y|X}(\boldsymbol{x})} V^n(\boldsymbol{y}|\boldsymbol{x}) \leq \exp\left\{-nD(P_{XY}||P_X \times V)\right\}, \tag{4.41}$$

where $P_X \times V$ denotes the distribution on $\mathcal{X} \times \mathcal{Y}$ with pmf $P_X(x)V(y|x)$ and $V^n(\boldsymbol{y}|\boldsymbol{x}) \triangleq \prod_{t=1}^n V(y_t|x_t)$.

The set of codewords $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\}$ used in proving the achievability result is any set with the properties stated in Lemma 4.10. It is shown in [45, Appendix] that a randomly chosen codeword set have these properties with probability arbitrarily close to 1.

Lemma 4.10 ([45, Lemma 3])**.** *For any $\epsilon > 0$, $n \geq n_0(\epsilon)$, $K \geq \exp(n\epsilon)$, and type P, there exist codewords $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\}$ in $\mathcal{X}^n$, each of type P, such that for every $\boldsymbol{x} \in \mathcal{X}^n$, $\boldsymbol{s} \in \mathcal{S}^n$, and every joint type $P_{XX'S}$, by setting $R = \frac{1}{n}\log K$, we have*

$$|\{j : (\boldsymbol{x}, \boldsymbol{x}_j, \boldsymbol{s}) \in \mathsf{T}_{XX'S}\}| \leq \exp\left\{n\left(|R - I(X';XS)|^+ + \epsilon\right)\right\}, \tag{4.42}$$

$$\frac{1}{K}|\{i : (\boldsymbol{x}_i, \boldsymbol{s}) \in \mathsf{T}_{XS}\}| \leq \exp(-n\epsilon/2), \qquad \text{if } I(X;S) > \epsilon, \tag{4.43}$$

and

$$\frac{1}{K}|\{i : (\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{s}) \in \mathsf{T}_{XX'S} \text{ for some } j \neq i\}| \leq \exp(-n\epsilon/2) \quad \text{if } I(X;X'S) - |R - I(X';S)|^+ > \epsilon. \tag{4.44}$$

In addition to Lemma 4.10, we need Lemma 4.11 (which is similar to [45, Lemma 4]), in order to establish the inambiguity of the decoding rule given in Definition 4.4.

Lemma 4.11. *If the PAVC is non-symmetrizable and $\beta > 0$, then for a sufficiently small η, no set of random variables X, X', S, S', Y can simultaneously satisfy*

$$P_X = P_{X'} = P \qquad \text{with} \qquad \min_{x \in \mathcal{X}} P(x) \geq \beta, \tag{4.45}$$

$$P_{XSY} \in \mathscr{D}_\eta, \qquad P_{X'S'Y} \in \mathscr{D}_\eta, \tag{4.46}$$

and

$$I(XY;X'|S) \leq \eta, \qquad I(X'Y;X|S') \leq \eta. \tag{4.47}$$

Proof. The proof technique is very similar to the proof of [45, Lemma 4]. □

So assuming that the decoder ϕ is being used as defined in Definition 4.4, lemma 4.11 proves that this decoder is unambiguously defined if η is chosen sufficiently small. In fact, if for some $\boldsymbol{y} \in \mathcal{Y}^n$ and some $i \neq j$, both $\boldsymbol{x}_i$ and

$\boldsymbol{x}_j$ satisfied conditions (1) and (2) in Definition 4.4, then some $\boldsymbol{s}$ and $\boldsymbol{s}'$ would exist, with the joint types of $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{s}, \boldsymbol{s}', \boldsymbol{y})$ represented by the dummy random variables X, X', S, S', Y (i.e., $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{s}, \boldsymbol{s}', \boldsymbol{y}) \in \mathsf{T}_{XX'SS'Y}$) that satisfy conditions stated in Lemma 4.11. This is in contradiction with Lemma 4.11.

The following lemma, Lemma 4.12, provides the error analysis for the decoder given in Definition 4.4.

Lemma 4.12. *Given any non-symmetrizable PAVC and arbitrary $\beta > 0$, $\delta > 0$, for any block length $n \geq n_0$ and any type P with $\min_x P(x) > \beta$, there exists a code with codewords $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\}$, each of type P, such that*

$$\frac{1}{n} \log K > I(P) - \delta, \tag{4.48}$$

and

$$\begin{aligned} \max_{P_{\boldsymbol{S}|\mathbf{q}(\boldsymbol{S})}} \mathbb{E}\left[\bar{e}_d(\boldsymbol{S})\right] &= \max_{P_{\boldsymbol{S}|\mathbf{q}(\boldsymbol{S})}} \sum_{\boldsymbol{s} \in \mathcal{S}^n} \bar{e}_d(\boldsymbol{s}) P_{\boldsymbol{S}|\mathbf{q}(\boldsymbol{S})}(\boldsymbol{s}|\mathbf{q}(\boldsymbol{s})) P_{Q^n}(\mathbf{q}(\boldsymbol{s})) \\ &< \exp(-n\gamma). \end{aligned} \tag{4.49}$$

Here, n_0 and $\gamma > 0$ depend only on the given PAVC, and on β and δ.

Proof. Let $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\}$ be as in Lemma 4.10, with $R = \frac{1}{n} \log K$ satisfying

$$I(P) - \delta < R < I(P) - \frac{2}{3}\delta, \tag{4.50}$$

and with ϵ (from Lemma 4.10) to be specified later. Let the decoder ϕ be as defined in Definition 4.4. Lemma 4.11 proves that this decoder ϕ is unambiguously defined if η is chosen sufficiently small.

To bound the decoding error, let us fix $P_{\boldsymbol{S}|\mathbf{q}(\boldsymbol{S})}$ and write

$$\begin{aligned} \mathbb{E}\left[\bar{e}_d(\boldsymbol{S})\right] &= \sum_{\boldsymbol{s} \in \mathcal{S}^n} \bar{e}_d(\boldsymbol{s}) P_{\boldsymbol{S}|\mathbf{q}(\boldsymbol{S})}(\boldsymbol{s}|\mathbf{q}(\boldsymbol{s})) P_{Q^n}(\mathbf{q}(\boldsymbol{s})) \\ &= \sum_{\boldsymbol{s} \in \mathcal{S}^n} \frac{1}{K} \sum_{i=1}^{K} \sum_{\boldsymbol{y}:\ \phi(\boldsymbol{y}) \neq i} W^n(\boldsymbol{y}|\boldsymbol{x}_i; \boldsymbol{s}) P_{\boldsymbol{S}|\mathbf{q}(\boldsymbol{S})}(\boldsymbol{s}|\mathbf{q}(\boldsymbol{s})) P_{Q^n}(\mathbf{q}(\boldsymbol{s})) \\ &= \sum_{\mathsf{T}_{\hat{S}}} \sum_{\boldsymbol{s} \in \mathsf{T}_{\hat{S}}} P_{\boldsymbol{S}|\mathbf{q}(\boldsymbol{S})}(\boldsymbol{s}|\mathbf{q}(\boldsymbol{s})) P_{Q^n}(\mathbf{q}(\boldsymbol{s})) \underbrace{\left(\frac{1}{K} \sum_{i=1}^{K} \sum_{\boldsymbol{y}:\ \phi(\boldsymbol{y}) \neq i} W^n(\boldsymbol{y}|\boldsymbol{x}_i; \boldsymbol{s}) \right)}_{\leq 1}. \end{aligned} \tag{4.51}$$

For $\eta \geq 0$, let us define a family of distribution P_S of random variables S with values from the set $\mathcal{S}$ by

$$\mathscr{S}_\eta \triangleq \left\{ P_S :\ D\left(P_S || P_Q \times P_{S|\mathbf{q}(S)}\right) \leq \eta \right\}, \tag{4.52}$$

where $P_{S|\mathfrak{q}(S)}$ is arbitrary and P_Q is the pmf over the channel classes of the PAVC, i.e., it is known and fixed. Then, by [46, Lemma 2.6, p.32], we may bound summation over $P_{\boldsymbol{S}|\mathfrak{q}(\boldsymbol{S})}(\boldsymbol{s}|\mathfrak{q}(\boldsymbol{s}))P_{Q^n}(\mathfrak{q}(\boldsymbol{s}))$ as follows

$$\begin{aligned}\sum_{\boldsymbol{s}\in\mathsf{T}_{\hat{S}}} P_{\boldsymbol{S}|\mathfrak{q}(\boldsymbol{S})}(\boldsymbol{s}|\mathfrak{q}(\boldsymbol{s}))P_{Q^n}(\mathfrak{q}(\boldsymbol{s})) &\leq \sum_{\boldsymbol{s}\in\mathsf{T}_{\hat{S}}} P_{Q^n}(\mathfrak{q}(\boldsymbol{s}))\\ &= P_{Q^n}(\mathsf{T}_{\hat{Q}})\\ &\leq \exp\left\{-nD(P_{\hat{Q}}||P_Q)\right\}, \end{aligned} \tag{4.53}$$

where $P_{\hat{Q}}$ is the distribution on $\mathfrak{q}(\hat{S})$ which is implied by $P_{\hat{S}}$. Now by Lemma 4.7, we have

$$\begin{aligned}&\mathbb{E}\left[\bar{e}_d(\boldsymbol{S})\right] \leq\\ &\leq \sum_{\substack{\mathsf{T}_{\hat{S}}:\\ P_{\hat{S}}\in\mathscr{S}_\eta}} \sum_{\boldsymbol{s}\in\mathsf{T}_{\hat{S}}} P_{\boldsymbol{S}|\mathfrak{q}(\boldsymbol{S})}(\boldsymbol{s}|\mathfrak{q}(\boldsymbol{s}))P_{Q^n}(\mathfrak{q}(\boldsymbol{s})) \underbrace{\left(\frac{1}{K}\sum_{i=1}^{K}\sum_{\boldsymbol{y}:\ \phi(\boldsymbol{y})\neq i} W^n(\boldsymbol{y}|\boldsymbol{x}_i;\boldsymbol{s})\right)}_{\bar{e}_d(\boldsymbol{s})}\\ &\quad + \exp\left(-n\frac{\eta}{2}\right). \end{aligned} \tag{4.54}$$

The rest of the proof is similar to that of [45, Lemma 5]. By fixing $\boldsymbol{s}$ such that $P_{\boldsymbol{s}} \in \mathscr{S}_\eta$ and following similar steps stated in [45, Lemma 5], we may bound the inner term in front of summation in the above expression and show that it is exponentially vanishing as $n \to \infty$. This in fact completes the proof of Lemma 4.12.

However, for completeness, we will state the rest of the proof as well. As we mentioned before, let us fix $\boldsymbol{s}$ such that $P_{\boldsymbol{s}} \in \mathscr{S}_\eta$ and observe that by (4.43) and Lemma 4.7 we have

$$\begin{aligned}\frac{1}{K}\left|\left\{i : (\boldsymbol{x}_i, \boldsymbol{s}) \in \bigcup_{I(X;S)>\epsilon} \mathsf{T}_{XS}\right\}\right| &\leq (\text{number of joint types})\cdot\exp(-n\epsilon/2)\\ &\leq \exp(-n\epsilon/3), \end{aligned} \tag{4.55}$$

for n larger than a suitable threshold n_0, that depends on ϵ.

So, in order to obtain an exponentially decreasing upper bound on $\bar{e}_d(\boldsymbol{s})$ (for those $\boldsymbol{s}$ such that $P_{\boldsymbol{s}} \in \mathscr{S}_\eta$), it is sufficient to consider only those codewords $\boldsymbol{x}_i$ for which $(\boldsymbol{x}_i, \boldsymbol{s}) \in \mathsf{T}_{XS}$ with $I(X;S) \leq \epsilon$. Then, for $P_{XSY} \notin \mathscr{D}_\eta$ (see (4.22)), we have

$$\begin{aligned}D(P_{XSY}||P_{XS}\times W) &= D(P_{XSY}||P_X\times P_Q\times P_{S|\mathfrak{q}(S)}\times W) - I(X;S)\\ &> \eta - \epsilon, \end{aligned} \tag{4.56}$$

and thus by Lemma 4.9, we can write

$$\sum_{\boldsymbol{y}\in\mathsf{T}_{Y|XS}(\boldsymbol{x}_i,\boldsymbol{s})} W^n(\boldsymbol{y}|\boldsymbol{x}_i;\boldsymbol{s}) \le \exp\{-D(P_{XSY}||P_{XS}\times W)\} \le \exp\{-n(\eta-\epsilon)\}. \tag{4.57}$$

Hence by Lemma 4.7, we have

$$\sum_{\boldsymbol{y}:\ P_{\boldsymbol{x}_i,\boldsymbol{s},\boldsymbol{y}}\notin\mathscr{D}_\eta} W^n(\boldsymbol{y}|\boldsymbol{x}_i;\boldsymbol{s}) \le \exp\{-n(\eta-2\epsilon)\}. \tag{4.58}$$

Next, note that if $P_{\boldsymbol{x}_i,\boldsymbol{s},\boldsymbol{y}}\in\mathscr{D}_\eta$ and $\phi(\boldsymbol{y})\neq i$, then condition (2) of Definition 4.4 must be violated. So let us denote by $\mathscr{E}_\eta$ the set of all joint distributions $P_{XX'SY}$ such that (i) $P_{XSY}\in\mathscr{D}_\eta$; (ii) $P_{X'S'Y}\in\mathscr{D}_\eta$ for some S'; and (iii) $I(XY;X'|S)>\eta$. Then, it follows that

$$\sum_{\substack{\boldsymbol{y}:\ P_{\boldsymbol{x}_i,\boldsymbol{s},\boldsymbol{y}}\in\mathscr{D}_\eta\\ \phi(\boldsymbol{y})\neq i}} W^n(\boldsymbol{y}|\boldsymbol{x}_i;\boldsymbol{s}) \le \sum_{P_{XX'SY}\in\mathscr{E}_\eta} e_{XX'SY}(i,\boldsymbol{s}), \tag{4.59}$$

where

$$e_{XX'SY}(i,\boldsymbol{s}) \triangleq \sum_{\substack{\boldsymbol{y}:\ (\boldsymbol{x}_i,\boldsymbol{x}_j,\boldsymbol{s},\boldsymbol{y})\in\mathsf{T}_{XX'SY}\\ \text{for some } j\neq i}} W^n(\boldsymbol{y}|\boldsymbol{x}_i;\boldsymbol{s}), \tag{4.60}$$

and the summation (4.59) extends to all joint types $P_{XX'SY}\in\mathscr{E}_\eta$ (of course, $e_{XX'SY}(i,\boldsymbol{s})=0$ unless $P_{X'}=P_X=P$ and $P_{XS}=P_{\boldsymbol{x}_i,\boldsymbol{s}}$).

Combining (4.55)-(4.59), for those $\boldsymbol{s}$ such that $P_{\boldsymbol{s}}\in\mathscr{S}_\eta$, we obtain that

$$\bar{e}_d(\boldsymbol{s}) \le \exp\{-n\epsilon/3\} + \exp\{-n(\eta-2\epsilon)\} + \frac{1}{K}\sum_{i=1}^{K}\sum_{P_{XX'SY}\in\mathscr{E}_\eta} e_{XX'SY}(i,\boldsymbol{s}). \tag{4.61}$$

Before finding an upper bound for $e_{XX'SY}(i,\boldsymbol{s})$, note that it is sufficient to do so only when $P_{XX'SY}\in\mathscr{E}_\eta$ satisfies

$$I(X;X'S) \le |R-I(X';S)|^+ + \epsilon, \tag{4.62}$$

otherwise, by (4.44), we have

$$\frac{1}{K}\left|\{i:(\boldsymbol{x}_i,\boldsymbol{x}_j,\boldsymbol{s})\in\mathsf{T}_{XX'S}\ \text{ for some } j\neq i\}\right| < \exp\{-n\epsilon/2\}. \tag{4.63}$$

Since $(\boldsymbol{x}_i,\boldsymbol{x}_j,\boldsymbol{s})\in\mathsf{T}_{XX'S}$ for some $j\neq i$ is a necessary condition for $e_{XX'SY}(i,\boldsymbol{s})>0$ (see (4.60)), it follows from Lemma 4.7 that the contribution to the double summation in (4.61) of the terms with $P_{XX'SY}\in\mathscr{E}_\eta$ not satisfying (4.62) is less than $\exp\{-n\epsilon/3\}$.

Now, from (4.60), we can write

$$e_{XX'SY}(i,\boldsymbol{s}) \leq \sum_{j:(\boldsymbol{x}_i,\boldsymbol{x}_j,\boldsymbol{s})\in\mathsf{T}_{XX'S}} \sum_{\boldsymbol{y}\in\mathsf{T}_{Y|XX'S}(\boldsymbol{x}_i,\boldsymbol{x}_j,\boldsymbol{s})} W^n(\boldsymbol{y}|\boldsymbol{x}_i;\boldsymbol{s}). \tag{4.64}$$

Because $W^n(\boldsymbol{y}|\boldsymbol{x}_i;\boldsymbol{s})$ is constant for $\boldsymbol{y}\in\mathsf{T}_{Y|XS}(\boldsymbol{x}_i,\boldsymbol{s})$ and this constant is less than or equal to $(|\mathsf{T}_{Y|XS}(\boldsymbol{x}_i,\boldsymbol{s})|)^{-1}$, the inner sum in (4.64) is bounded above by

$$\left|\mathsf{T}_{Y|XX'S}(\boldsymbol{x}_i,\boldsymbol{x}_j,\boldsymbol{s})\right| \cdot \left(\left|\mathsf{T}_{Y|XS}(\boldsymbol{x}_i,\boldsymbol{s})\right|\right)^{-1}, \tag{4.65}$$

which in turn, by Lemma 4.8, is less than or equal to $\exp\{-n[I(Y;X'|XS)-\epsilon]\}$. Now by using (4.42), it follows from (4.64) that

$$e_{XX'SY}(i,\boldsymbol{s}) \leq \exp\left\{-n\left[I(Y;X'|XS) - |R-I(X';XS)|^+ - 2\epsilon\right]\right\}. \tag{4.66}$$

In order to further bound $e_{XX'SY}(i,\boldsymbol{s})$ when (4.62) holds, we distinguish between two cases: a) $R\leq I(X';S)$, and b) $R>I(X';S)$.

For the case a), from (4.62) we have

$$I(X;X'|S) \leq I(X;X'S) \leq \epsilon, \tag{4.67}$$

and hence by condition (iii) in the definition of $\mathscr{E}_\eta$, we can write

$$I(Y;X'|XS) = I(XY;X'|S) - I(X;X'|S) \geq \eta - \epsilon. \tag{4.68}$$

Since for this case we have $R\leq I(X';S)\leq I(X';XS)$, it follows from (4.66) that

$$e_{XX'SY}(i,\boldsymbol{s}) \leq \exp\{-n(\eta-3\epsilon)\}. \tag{4.69}$$

In case b), from (4.62) we have

$$\begin{aligned} R &> I(X;X'S) + I(X';S) - \epsilon \\ &= I(X';XS) + I(X;S) - \epsilon \\ &\geq I(X';XS) - \epsilon, \end{aligned} \tag{4.70}$$

and hence

$$|R-I(X';XS)|^+ \leq R - I(X';XS) + \epsilon. \tag{4.71}$$

Substituting this into (4.66) it follows that

$$\begin{aligned} e_{XX'SY}(i,\boldsymbol{s}) &\leq \exp\left\{-n\left[I(X';XSY) - R - 3\epsilon\right]\right\} \\ &\leq \exp\left\{-n\left[I(X';Y) - R - 3\epsilon\right]\right\}. \end{aligned} \tag{4.72}$$

Note that $P_{XX'SY}\in\mathscr{E}_\eta$ implies that $P_{X'S'Y}\in\mathscr{D}_\eta$ for some S'. So by definition of $\mathscr{D}_\eta$ given in (4.22), $P_{X'S'Y}$ is arbitrary close to $P_{X''S''Y''}\in\mathscr{D}_0$ defined by $P_{X''S''Y''} = P\times P_Q \times P_{S'|\mathfrak{q}(S')}\times W$. Now if η is sufficiently small, then $I(X';Y)$ is arbitrarily close to $I(X'';Y'')$, say, $I(X';Y)\geq I(X'';Y'')-\delta/3$.

Using the definition of $I(P)$ given in (4.23) and the assumption (4.50), we can write

$$I(X';Y) - R \geq I(X'';Y'') - \delta/3 - R \geq I(P) - \delta/3 - R \geq \delta/3, \tag{4.73}$$

if η is sufficiently small and depends only on δ. Fixing η accordingly and also small enough for the decoding rule to be unambiguous, (4.72) yields for case b) that

$$e_{XX'SY}(i, \boldsymbol{s}) \leq \exp\left\{-n\left[\frac{\delta}{3} - 3\epsilon\right]\right\}. \tag{4.74}$$

Now, from (4.61), by using (4.69) and (4.74) and Lemma 4.7, we obtain that

$$\bar{e}_d(\boldsymbol{s}) \leq \exp(-n\epsilon/4), \tag{4.75}$$

if, for instance, $\epsilon \leq \min[\eta/4, \delta/10]$ and n is sufficiently large. Because the bound holds uniformly for those $\boldsymbol{s}$ such that $P_{\boldsymbol{s}} \in \mathscr{S}_\eta$, then by substituting it into (4.54) and using Lemma 4.7, the proof of Lemma 4.12 becomes complete. □

4.B Capacity of a PAVC with Stochastic Encoder: Proof of Theorem 4.2

Proof of Theorem 4.2. Because deterministic codes are special cases of codes with stochastic encoder, the achievability part of this theorem directly follows from that of Theorem 4.1.

The converse part of the theorem follows from similar steps that have been used in the proof of Theorem 4.1, i.e., Lemma 4.5 and Lemma 4.6.

When the rate is greater than $I(P)$, defined in (4.23), the converse proof follows from the converse proof of randomized codes, i.e., Lemma 4.14, by choosing the random decoder Φ to be a fixed decoder ϕ (this does not change any part of the proof). When the channel is symmetrizable, the converse follows from Lemma 4.13 and this completes the proof. □

Lemma 4.13. *For a symmetrizable PAVC, any stochastic code of block length n with $K \geq 2$ codewords, each of type P has*

$$\mathbb{E}\left[\bar{e}_d(\boldsymbol{S})\right] = \max_{P_{\boldsymbol{S}|\mathbf{q}(\boldsymbol{S})}} \sum_{\boldsymbol{s}\in\mathcal{S}^n} \bar{e}_d(\boldsymbol{s}) P_{\boldsymbol{S}|\mathbf{q}(\boldsymbol{S})}(\boldsymbol{s}|\mathbf{q}(\boldsymbol{s})) P_{Q^n}(\mathbf{q}(\boldsymbol{s})) \geq \frac{1}{4}. \tag{4.76}$$

Proof. Consider an arbitrary stochastic code (Ψ, ϕ) which is defined over the message set $\mathcal{M} = \{1, \ldots, K\}$. Let the random variable Ψ be defined over a set of L encoders $\left\{\psi^{(1)}, \ldots, \psi^{(L)}\right\}$ with a pmf P_Ψ where $P_\Psi(l)$ is the probability of choosing the lth encoder $\psi^{(l)}$.

For some $U \in \mathcal{U}(\mathcal{X} \times \mathcal{Q} \to \mathcal{S})$ satisfying (4.14) consider K random sequences $\boldsymbol{S}_j = (S_{j1}, \ldots, S_{jn})$ where $\boldsymbol{S}_j \in \mathcal{S}^n$, $j \in [1 : K]$, is chosen according to the

following distribution

$$\begin{aligned}\mathbb{P}\left[\boldsymbol{S}_j=\boldsymbol{s}\right] &= \sum_{l=1}^{L}\left[\prod_{k=1}^{n} U(s_k|\psi^{(l)}(j)_k,\mathfrak{q}(s_k))P_Q(\mathfrak{q}(s_k))\right]P_\Psi(l)\\ &= \left[\sum_{l=1}^{L}\prod_{k=1}^{n} U(s_k|\psi^{(l)}(j)_k,\mathfrak{q}(s_k))P_\Psi(l)\right]\left[\prod_{k'=1}^{n} P_Q(\mathfrak{q}(s_{k'}))\right]\\ &= \underbrace{\left[\sum_{l=1}^{L}\prod_{k=1}^{n} U(s_k|\psi^{(l)}(j)_k,\mathfrak{q}(s_k))P_\Psi(l)\right]}_{P_{\boldsymbol{S}|\mathfrak{q}(\boldsymbol{S})}} P_{Q^n}(\mathfrak{q}(\boldsymbol{s})). \end{aligned} \tag{4.77}$$

Then for each pair (i,j) and every $\boldsymbol{y}=(y_1,\ldots,y_n)\in\mathcal{Y}^n$ we can write

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{S}_j}\left[\mathbb{E}_\Psi\left[W^n(\boldsymbol{y}|\Psi(i);\boldsymbol{S}_j)\right]\right] =\\ &= \mathbb{E}_\Psi\left[\sum_{\boldsymbol{s}\in\mathcal{S}^n}\left[\prod_{k=1}^{n} W(y_k|\Psi(i)_k;s_k)\right]\mathbb{P}\left[\boldsymbol{S}_j=\boldsymbol{s}\right]\right]\\ &= \mathbb{E}_\Psi\left[\sum_{l=1}^{L}\left[\sum_{\boldsymbol{s}\in\mathcal{S}^n}\prod_{k=1}^{n} W(y_k|\Psi(i)_k;s_k)U(s_k|\psi^{(l)}(j)_k,\mathfrak{q}(s_k))P_Q(\mathfrak{q}(s_k))\right]P_\Psi(l)\right]\\ &= \sum_{l'=1}^{L}\sum_{l=1}^{L}\left[\sum_{\boldsymbol{s}\in\mathcal{S}^n}\prod_{k=1}^{n} W(y_k|\psi^{(l')}(i)_k;s_k)U(s_k|\psi^{(l)}(j)_k,\mathfrak{q}(s_k))P_Q(\mathfrak{q}(s_k))\right]P_\Psi(l)P_\Psi(l')\\ &= \sum_{l'=1}^{L}\sum_{l=1}^{L}\left[\prod_{k=1}^{n}\sum_{s\in S} W(y_k|\psi^{(l')}(i)_k;s)U(s|\psi^{(l)}(j)_k,\mathfrak{q}(s))P_Q(\mathfrak{q}(s))\right]P_\Psi(l)P_\Psi(l'). \end{aligned} \tag{4.78}$$

So, by using (4.14), it follows that

$$\mathbb{E}_{\boldsymbol{S}_j}\left[\mathbb{E}_\Psi\left[W^n(\boldsymbol{y}|\Psi(i);\boldsymbol{S}_j)\right]\right] = \mathbb{E}_{\boldsymbol{S}_i}\left[\mathbb{E}_\Psi\left[W^n(\boldsymbol{y}|\Psi(j);\boldsymbol{S}_i)\right]\right], \tag{4.79}$$

and hence for $i\neq j$ we have

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{S}_j}\left[e_t(i,\boldsymbol{S}_j)\right]+\mathbb{E}_{\boldsymbol{S}_i}\left[e_t(j,\boldsymbol{S}_i)\right] =\\ &= \sum_{\boldsymbol{y}:\ \phi(\boldsymbol{y})\neq i}\mathbb{E}_{\boldsymbol{S}_j}\left[\mathbb{E}_\Psi\left[W^n(\boldsymbol{y}|\Psi(i);\boldsymbol{S}_j)\right]\right]+\sum_{\boldsymbol{y}:\ \phi(\boldsymbol{y})\neq j}\mathbb{E}_{\boldsymbol{S}_i}\left[\mathbb{E}_\Psi\left[W^n(\boldsymbol{y}|\Psi(j);\boldsymbol{S}_i)\right]\right]\\ &\geq \sum_{\boldsymbol{y}\in\mathcal{Y}^n}\mathbb{E}_{\boldsymbol{S}_j}\left[\mathbb{E}_\Psi\left[W^n(\boldsymbol{y}|\Psi(i);\boldsymbol{S}_j)\right]\right]\\ &= 1. \end{aligned} \tag{4.80}$$

Now, from here on the proof is very similar to that of Lemma 4.5. Using the

above fact we can write

$$\begin{aligned}\frac{1}{K}\sum_{j=1}^{K}\mathbb{E}_{\boldsymbol{S}_j}\left[\bar{e}_t(\boldsymbol{S}_j)\right] &= \frac{1}{K^2}\sum_{i=1}^{K}\sum_{j=1}^{K}\mathbb{E}_{\boldsymbol{S}_j}\left[e_t(i,\boldsymbol{S}_j)\right]\\ &\geq \frac{1}{K^2}\cdot\frac{K(K-1)}{2}\\ &= \frac{K-1}{2K},\end{aligned} \tag{4.81}$$

so it follows that for some $j \in [1:K]$ we have

$$\mathbb{E}_{\boldsymbol{S}_j}\left[\bar{e}_t(\boldsymbol{S}_j)\right] \geq \frac{K-1}{2K} \geq \frac{1}{4}. \tag{4.82}$$

This leads to the desired result because $\mathbb{E}\left[\bar{e}_t(\boldsymbol{S})\right] \geq 1/4$ for some distribution over $\boldsymbol{S}$ of the form $P_{\boldsymbol{S}|\mathfrak{q}(\boldsymbol{S})}P_{Q^n}$ where $P_{\boldsymbol{S}|\mathfrak{q}(\boldsymbol{S})}$ is given in (4.77). So in general we have $\max_{P_{\boldsymbol{S}|\mathfrak{q}(\boldsymbol{S})}}\mathbb{E}\left[\bar{e}_d(\boldsymbol{S})\right] \geq 1/4$ and we are done. □

4.C Randomized Code Capacity of a PAVC: Proof of Theorem 4.3

Suppose that there are k non-negative-valued functions $l_1,\ldots,l_k$ on $\mathcal{S}$ where for simplicity we assume that $\min_{s\in\mathcal{S}} l_i(s) = 0$. Given $\Lambda_1,\ldots,\Lambda_k$, we say that $\boldsymbol{s} \in \mathcal{S}^n$ satisfies state constraints $\Lambda_1,\ldots,\Lambda_k$, if $l_i(\boldsymbol{s}) \leq \Lambda_i$ for all i, where

$$l(\boldsymbol{s}) = \frac{1}{n}\sum_{t=1}^{n} l(s_t), \qquad \boldsymbol{s} \in \mathcal{S}^n. \tag{4.83}$$

By applying the same method of [47], the result of [47, Theorem 3.1] can be extended to multiple state constraints as stated in the following result.

Theorem 4.5. *The* randomized *code capacity of the AVC* (4.3) *under state constraint* $\Lambda_1,\ldots,\Lambda_k$*, denoted by* $C_{\mathsf{avc}}^{\mathsf{r}}(\Lambda)$*, is determined in [47], and is given by*

$$\begin{aligned}C_{\mathsf{avc}}^{\mathsf{r}}(\Lambda_1,\ldots,\Lambda_k) &= \max_{P_X}\min_{P_S:\forall i\ \mathbb{E}[l_i(S)]\leq\Lambda_i} I(P_X,\bar{W}_S)\\ &= \min_{P_S:\forall i\ \mathbb{E}[l_i(S)]\leq\Lambda_i}\max_{P_X} I(P_X,\bar{W}_S).\end{aligned} \tag{4.84}$$

Proof of Theorem 4.3. The converse part, using a similar argument to [47, Lemma 3.2 and Theorem 3.1], follows from Lemma 4.14. In the following we prove the achievability part.

Define an AVC with the following convergent state constraints. For each $i \in \mathcal{Q}$, define a non-negative-valued function l_i on $\boldsymbol{s} \in \mathcal{S}^n$ as

$$l_i(\boldsymbol{s}) \triangleq \frac{1}{n}\sum_{t=1}^{n}\mathbb{1}_{\mathfrak{q}(s_t)=i}. \tag{4.85}$$

For any $\epsilon > 0$, consider the state constraints

$$|l_i(\boldsymbol{s}) - P_Q(i)| \le \epsilon, \forall i \in \mathcal{Q}. \tag{4.86}$$

By Theorem 4.5, the capacity of the AVC under the state constraints (4.86) is

$$\begin{aligned} C_{\mathsf{avc}}^{\mathsf{r}}(P_Q, \epsilon) &\triangleq \max_{P_X} \min_{\substack{P_S: \\ \forall i \in \mathcal{Q},\ |\mathbb{P}[\mathfrak{q}(S)=i] - P_Q(i)| \le \epsilon}} I(P_X, \bar{W}_S) \\ &= \min_{\substack{P_S: \\ \forall i \in \mathcal{Q},\ |\mathbb{P}[\mathfrak{q}(S)=i] - P_Q(i)| \le \epsilon}} \max_{P_X} I(P_X, \bar{W}_S), \end{aligned} \tag{4.87}$$

where we use $\mathbb{E}[l_i(S)] = \mathbb{P}[\mathfrak{q}(S) = i]$. By the monotonicity and the continuity of $C_{\mathsf{avc}}^{\mathsf{r}}(P_Q, \epsilon)$ as a function of ϵ,

$$C_{\mathsf{pavc}}^{\mathsf{r}} = \sup_{\epsilon > 0} C_{\mathsf{avc}}^{\mathsf{r}}(P_Q, \epsilon). \tag{4.88}$$

Then we show that any rate $\mathfrak{R} < C_{\mathsf{pavc}}^{\mathsf{r}} = \sup_{\epsilon>0} C_{\mathsf{avc}}^{\mathsf{r}}(P_Q, \epsilon)$ is achievable for PAVC.

Pick an ϵ_0 such that $\mathfrak{R} < C_{\mathsf{avc}}^{\mathsf{r}}(P_Q, \epsilon_0)$, which is possible by (4.88). Fix any $\varepsilon > 0$ and $\delta > 0$. Choose ε' with $0 < \varepsilon' < \varepsilon$. Since $\mathfrak{R}$ is achievable for the AVC with the state constraints (4.86), with ε' in place of ϵ and for sufficiently large n, there exists a random code (Ψ, Φ) of blocklength n, rate larger than $\mathfrak{R} - \delta$ and

$$\bar{e}_r(\boldsymbol{s}) \le \varepsilon' \tag{4.89}$$

for all state sequences satisfying (4.86) with ε' in place of ϵ. For a random sequence $\boldsymbol{S}$ of PAVC, by Hoeffding's inequality,

$$\mathbb{P}[|l_i(\boldsymbol{S}) - P_Q(i)| \le \epsilon_0, \forall i \in \mathcal{Q}] \ge 1 - 2\exp(-2\epsilon_0^2 n). \tag{4.90}$$

For random code (Ψ, Φ) with sufficiently large n such that $2\exp(-2\epsilon_0^2 n) < \varepsilon - \varepsilon'$, we have

$$\begin{aligned} \mathbb{E}[\bar{e}_r(\boldsymbol{S})] &\le \mathbb{E}[\bar{e}_r(\boldsymbol{S})|\ |l_i(\boldsymbol{S}) - P_Q(i)| \le \epsilon_0, \forall i \in \mathcal{Q}] \\ &\quad + \mathbb{P}[|l_i(\boldsymbol{S}) - P_Q(i)| > \epsilon_0, \text{ for some } i \in \mathcal{Q}] \\ &< \varepsilon' + \varepsilon - \varepsilon'. \end{aligned} \tag{4.91}$$

Thus for sufficiently large n, there exists blocklength n random code for PAVC with rate larger than $\mathfrak{R} - \delta$ and $\mathbb{E}[\bar{e}_r(S)] < \varepsilon$. Therefore, $\mathfrak{R}$ is achievable for PAVC. This completes the proof of the theorem. □

Lemma 4.14. *For any $\delta > 0$ and $\epsilon < 1$, there exists n_0 such that for any randomized code (Ψ, Φ) of block length $n \ge n_0$, having*

$$\frac{1}{n} \log K \ge \min_{P_{S|\mathfrak{q}(S)}} \max_{P_X} I(P_X, \bar{W}_S) + \delta \tag{4.92}$$

implies

$$\mathbb{E}[\bar{e}_r(\boldsymbol{S})] = \max_{P_{\boldsymbol{S}|\mathfrak{q}(\boldsymbol{S})}} \sum_{\boldsymbol{s} \in \mathcal{S}^n} \bar{e}_r(\boldsymbol{s}) P_{\boldsymbol{S}|\mathfrak{q}(\boldsymbol{S})}(\boldsymbol{s}|\mathfrak{q}(\boldsymbol{s})) P_{Q^n}(\mathfrak{q}(\boldsymbol{s})) > \epsilon. \tag{4.93}$$

Proof. Let us fix $P_{S|\mathfrak{q}(S)}$ and assume that $P_X = P^*$ achieves the maximum of $I(P_X, \bar{W}_S)$ for this choice. Now, let $\boldsymbol{S} = (S_1, \ldots, S_n)$ be n independent realization of S according to the distribution $P_{S|\mathfrak{q}(S)}P_Q$. Then we can write

$$\begin{aligned}
\mathbb{E}\left[\bar{e}_r(\boldsymbol{S})\right] &= \frac{1}{K}\sum_{i=1}^{K}\mathbb{E}\left[e_r(i,\boldsymbol{S})\right] \\
&= \frac{1}{K}\sum_{i=1}^{K}\mathbb{E}_{\boldsymbol{S}}\left[\mathbb{E}_{\Psi,\Phi}\left[e(i,\boldsymbol{S},\Psi,\Phi)\right]\right] \\
&= \frac{1}{K}\sum_{i=1}^{K}\mathbb{E}_{\Psi,\Phi}\left[\sum_{\boldsymbol{y}:\Phi(\boldsymbol{y})\neq i}\mathbb{E}_{\boldsymbol{S}}\left[W^n(\boldsymbol{y}|\Psi(\boldsymbol{x});\boldsymbol{S})\right]\right] \\
&= \mathbb{E}_{\Psi,\Phi}\left[\frac{1}{K}\sum_{i=1}^{K}\sum_{\boldsymbol{y}:\Phi(\boldsymbol{y})\neq i}\prod_{j=1}^{n}\mathbb{E}_{\boldsymbol{S_j}}\left[W(y_j|\Psi(\boldsymbol{x})_j;S_j)\right]\right]. \qquad (4.94)
\end{aligned}$$

All of the random variables S_j are i.i.d., so if we introduce a new discrete memory-less channel (DMC) $\bar{W}_S$ defined by

$$\bar{W}_S(y|x) = \mathbb{E}\left[W(y|x;S)\right],$$

then we have

$$\begin{aligned}
\mathbb{E}\left[\bar{e}_r(\boldsymbol{S})\right] &= \mathbb{E}_{\Psi,\Phi}\left[\frac{1}{K}\sum_{i=1}^{K}\sum_{\boldsymbol{y}:\Phi(\boldsymbol{y})\neq i}\prod_{j=1}^{n}\bar{W}_S(y_j|\Psi(\boldsymbol{x})_j)\right], \\
&= \mathbb{E}_{\Psi,\Phi}\left[\bar{e}_{(\bar{W}_S)}(\Psi,\Phi)\right], \qquad (4.95)
\end{aligned}$$

where $\bar{e}_{(\bar{W}_S)}(\psi,\phi)$ is the average probability of error when a code (ψ,ϕ) is used on the DMC $\bar{W}_S$. Now, by using the strong converse to the coding theorem for the DMC $\bar{W}_S$, every code (ψ,ϕ) of rate $R \geq \max_{P_X} I(P_X,\bar{W}_S) + \delta$ has an average error probability $\bar{e}_{(\bar{W}_S)}(\psi,\phi)$ arbitrary close to 1 if n is large enough. So as a result, for every $\epsilon < 1$ we have $\mathbb{E}\left[\bar{e}_r(\boldsymbol{S})\right] > \epsilon$ and this completes the proof. □

"Everywhere is within walking distance if you have the time."

- Steven Wright

5 Compressed Network Coding Vectors

Practical networks being subject to random delays, synchronization errors, and even packet erasures, nodes failures, and topology changes, it is not viable to assume that the linear combinations performed at the intermediate nodes are deterministically known at the receivers.

Two approaches have been proposed in the literature to address this. The first has a coding vector appended to each packet [20]. This vector keeps track of the linear combination of the source packets the coded packet contains. The receivers use this information to solve a system of linear equations and recover the original data. The second approach uses subspace coding [1]. The information is conveyed by a subspace that the source selects; the receivers to decode simply need to decide which was the sent subspace. In this case the receiver needs no information about the linear combinations that the network nodes perform to decode. Both these approaches divide the source packets into generations and allow combining only among packets in the same generation. As far as we know, these are the only two approaches currently proposed.

The first approach comes at the cost of the coding vectors overhead. This overhead would be acceptable for large packets, however, in wireless applications, where packets are much shorter, it can very fast become prohibitive. Even in wired networks, the trade-off between a larger generation, which employs longer coding vectors, and a smaller generation, which may not allow mixing of packets and reduce the NC benefits, is a subject of research in the community.

From an information theoretic point of view discussed in Chapter 3, the second approach, subspace coding, results in higher information rates for short packet length (more precisely for small field size), but as the packet length increases, achieves the same information rate as having the coding vectors overhead.

In this chapter we present a third approach that is not a special case of the previous two. Our approach employs shortened or compressed coding vectors to efficiently convey the coding coefficients. The observation our approach leverages is that, the classic design of coding vectors allows potentially *all* source packets to get combined together; however, for some networks, this is too strong a requirement (see Section 5.1 for examples), and results in too low an information rate. In our approach we thus propose to employ coding vectors that allow at most m source packets to get combined. This naturally occurs in some applications, where for example only source packets originating from neighboring nodes get combined. We can also artificially restrict the number of source packets that get combined, by appending to each coded packet a few bits to count the number of source packets it contains. Note that, the receiver will eventually still need to solve a set of n linear equations to retrieve the source data; our approach only shortens the coding vectors that convey these linear combinations.

Our design problem can now be stated as follows. Given a generation that contains n source packets, each receiver is going to observe packets that contain linear combinations of *at most* m source packets. We want to design coding vectors that allow us, by receiving each combined packet, to determine which linear combination of the source packets it contains. The classical coding vectors design would utilize coding vectors of length n. In this chapter we explore what, under our assumptions, is the smallest length r of coding vectors we need to employ, and how can we select them. A key point of our design is that we require the intermediate node operation to be oblivious to the coding vectors employed, and in particular, to not perform compression operations.

For m much smaller than n, our approach can also be viewed as compressing the classical coding vectors, and our problem can be cast in a compressed sensing framework. Moreover, in this case, solving the set of n linear equations at the receiver becomes more efficient, since we can take advantage of the low density of the linear combinations to decode with belief propagation techniques.

It is important to mention that this chapter has been done as a joint work with Lorenzo Keller [1].

5.1 Problem Statement

Consider a dissemination protocol where the nodes in the network perform linear NC, i.e., linearly combine their incoming packets. One or multiple sources, not necessarily collocated, produce independent information packets, that we will call source packets. The source(s) packets get divided into sets called generations. Assume that each generation contains n source packets $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ where $\boldsymbol{x}_i \in \mathbb{F}_q^L$.

As we have discussed in Section 2.2, the classical coding vector approach appends to each source packet $\boldsymbol{x}_i$ a coding vector $\boldsymbol{x}_i^C$. Each receiver that receives

1. Lorenzo Keller is a Ph.D. student at Ecole Polytechnique Fédérale de Lausanne (EPFL), working under the supervision of prof. Christina Fragouli.

n packets $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n$, with linearly independent coding vectors can recover the original source information by solving the linear system of equations (see (2.18))

$$\begin{bmatrix} \boldsymbol{p}_1^I \\ \boldsymbol{p}_2^I \\ \vdots \\ \boldsymbol{p}_n^I \end{bmatrix} = \underbrace{\begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1n} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{n1} & \beta_{n2} & \cdots & \beta_n \end{bmatrix}}_{\boldsymbol{H} \in \mathbb{F}_q^{n \times n}} \begin{bmatrix} \boldsymbol{x}_1^I \\ \boldsymbol{x}_2^I \\ \vdots \\ \boldsymbol{x}_n^I \end{bmatrix}. \tag{5.1}$$

However, this approach comes at the overhead of the coding vectors, that can fast become impractical, as the following example illustrates.

Example 5.1. *Consider a sensor network consisting of* 100 *nodes, each sending a message to a sink. To implement NC using coding vectors over a field of size* $q = 2^4$ *we would need to use* 50 *Bytes of each packet simply for the coding vectors. In the TinyOs operating system [22], which is perhaps the most popular for sensor nodes, a typical frame length allows approximately* 30 *bytes for data transmissions. Thus clearly this is not a viable approach.*

As an alternative scheme, as explained in Section 2.2, subspace coding dispenses of the need to convey coding vectors. In this scheme, source(s) can only communicate information using subspaces which are unaffected by the linear operations performed on them. As it is shown in Chapter 3, this approach is optimal in terms of achievable information rates. When the length of the packets is small subspace coding results in higher transmission rate but as the length increases, essentially, it results in the same information rate as the coding vectors approach. Moreover, as the following example illustrates, code design is not trivial when multiple sources insert data in the network [2].

Example 5.2. *We here argue that designing subspace codes for the case where the sources are not collocated is challenging. Consider the case where n sources employ codebooks* $\mathcal{C}_i$, $i \in [1:n]$, *consisting of subspaces of the vector space* $\mathbb{F}_q^L$, *i.e.,*

$$\mathcal{C}_i = \left\{ \pi_j^{(i)} : \pi_j^{(i)} \sqsubseteq \mathbb{F}_q^L,\ 1 \leq j \leq |\mathcal{M}_i| \right\}, \quad i \in [1:n], \tag{5.2}$$

where $\mathcal{M}_i$ *is the message set for the ith source. To transmit information to the sink, source i maps a measured value to one such subspace* π *and inserts in the network* $\dim(\pi)$ *vectors that span* π. *In relaying information towards the sink, each sensor linearly combines all packets it has received (including that generated by itself) and transmits the combined packet to the next relays towards to the sink. As a result, the sink will observe vectors from the union of subspaces inserted by all the sources. In particular, if source i inserts the subspace* π_i, *the sink will observe vectors from the subspace* $\pi_1 + \pi_2 + \cdots + \pi_n$.

2. Note that the coding scheme introduced in Section 3.4 achieves the optimal performance asymptotically in the field size (packet length). However, when the field size is not large the problem of subspace code design for multiple sources become a hard problem.

Table 5.1 – An example of subspace coding for two sources.

$\mathcal{C}_2/\mathcal{C}_1$	π_1	π_2	π_3
π_4	$\pi_1 + \pi_4$	$\pi_2 + \pi_4$	$\pi_3 + \pi_4$
π_5	$\pi_1 + \pi_5$	$\pi_2 + \pi_5$	$\pi_3 + \pi_5$
π_6	$\pi_1 + \pi_6$	$\pi_2 + \pi_6$	$\pi_3 + \pi_6$

Using the knowledge of the codebooks $\{\mathcal{C}_i\}_{i=1}^n$, the sink needs to decode the source data.

To be able to correctly decode at the receiver, we need to ensure that every combination of source data results in a distinct *union subspace. We call this the* identifiability *property. Assume for simplicity we have two source nodes, S_1 using the codebook $\mathcal{C}_1 = \{\pi_1,\ \pi_2,\ \pi_3\}$, while S_2 using the codebook $\mathcal{C}_2 = \{\pi_4,\ \pi_5,\ \pi_6\}$. Table 5.1 summarizes all outcomes.*

For this code to be identifiable, we want all entries in Table 5.1 to correspond to distinct subspaces. For example, $\pi_1 + \pi_4$ should be a distinct subspace from $\pi_2 + \pi_5$.

This problem is hard to solve even for the case of two sources, and a very small codebook (in our example each node transmits only 3 values). Designing such a code for multiple sources is clearly a challenging task.

In both of the previous approaches, a common underlying assumption is that, all source packets may get combined in the network. Given that clearly this can be a too strong requirement for many practical networks, we here relax it, and require that each coded packets contains a linear combination of at most m out the n source packets. This allows us to use coding vectors whose length grows sub-linearly with n, resulting in a more efficient network communication. In the following we in turn discuss, how we can design such coding vectors, and how we utilize them in decoding, i.e., how we can retrieve the linear coefficients of the combined source packets. We also discuss what is the smallest required length, and what are the benefits we can expect to get.

5.2 Main Result

Here, we express the main result of this chapter as it is stated in Theorem 5.1.

Theorem 5.1. *Consider a NC scenario where each generation consists of n source(s) packets $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. Moreover, we assume that every packet $\boldsymbol{p}$ traversing the network does not contain any linear combinations of more than m of the original source(s) packets, where $m \leq n$. Then, if $m < \frac{n}{2}$, there exists an end-to-end coding scheme that result in shorter coding vectors. More specifically, if $m < \frac{n}{4}$ then the length of coding vectors behaves asymptotically as $O(m \log n)$ instead of n.*

In the next section, we will provide our coding scheme and prove the result of Theorem 5.1.

5.3 Compressing the Coding Vectors

In this section, we present our scheme for the compression of coding vectors.

5.3.1 Code Design

Consider a network performing linear NC, where each coded packet contains the linear combination of at most m source packets. For m much smaller than n, the classical coding vectors become sparse. We can thus compress them, by replacing them with shorter vectors, that still allow the receivers to extract the original coding vectors and decode the sources messages. Our construction utilizes properties of algebraic error correcting codes, and proceeds as follows.

Select a linear code $\mathcal{C} = [n, k, d]_q$ where $d = \min[2m + 1, n + 1]$ with k as large as possible. Consider the $r \times n$ parity check matrix $\mathbf{H}_\mathcal{C}$ where $r \triangleq n - k$. As coding vector, assign to source packet $\boldsymbol{x}_i$ the ith column of the matrix $\mathbf{H}_\mathcal{C}$, which we will denote as $\boldsymbol{h}_i$. That is,

$$\boldsymbol{h}_i = \boldsymbol{e}_i \cdot \mathbf{H}_\mathcal{C}^{\mathrm{T}}. \tag{5.3}$$

We call these vectors *compressed coding vectors*. Thus the sources insert to the network the packets

$$\boldsymbol{x}_i = [\boldsymbol{h}_i \mid \boldsymbol{x}_i^I]. \tag{5.4}$$

Intermediate nodes linearly combine their received packets. The coded packets propagating in the network will now have the form

$$\boldsymbol{p} \triangleq [\hat{\boldsymbol{p}}^C \mid \boldsymbol{p}^I], \tag{5.5}$$

where $\hat{\boldsymbol{p}}^C \in \mathbb{F}_q^r$ denotes the compressed coding vector appended to packet $\boldsymbol{p}$. This is related to the classical coding vector $\boldsymbol{p}^C$ that describes the linear transform from the source packets as

$$\hat{\boldsymbol{p}}^C = \boldsymbol{p}^C \cdot \mathbf{H}_\mathcal{C}^{\mathrm{T}}. \tag{5.6}$$

If m packets are allowed to be combined, with m much smaller than the length n of the coding vector $\boldsymbol{p}^C$, this can be viewed as compressing the sparse vector $\boldsymbol{p}^C$, and hence the compressed coding vector terminology.

The reason this construction enables receivers to decode follows from a well known property the columns of matrix $\mathbf{H}_\mathcal{C}$ satisfy. If a code $\mathcal{C}$ has minimum distance d, then any set of $d - 1$ columns of the matrix $\mathbf{H}_\mathcal{C}$ are linearly independent [48]. Moreover, given that at most m source packets get combined, $\mathrm{hwt}(\boldsymbol{p}^C) \leq m$ where $\mathrm{hwt}(\cdot)$ denotes the Hamming weight of a vector, the number of non-zero elements. The following lemma states that we will be able to recover the original coding vectors from the compressed ones.

Lemma 5.1. *There is an injective map between* $\boldsymbol{p}^C$, $\mathrm{hwt}(\boldsymbol{p}^C) \leq m$, *and* $\hat{\boldsymbol{p}}^C$ *related by (5.6).*

Proof. For two $\boldsymbol{p}_1^C \neq \boldsymbol{p}_2^C$ where $\mathrm{hwt}(\boldsymbol{p}_1^C) \leq m$ and $\mathrm{hwt}(\boldsymbol{p}_2^C) \leq m$ we have $\mathrm{hwt}(\boldsymbol{p}_2^C - \boldsymbol{p}_1^C) \leq \min[2m, n]$. But the minimum distance of $\mathbf{H}_{\mathcal{C}}$ is $\min[2m + 1, n + 1]$ so $(\boldsymbol{p}_2^C - \boldsymbol{p}_1^C) \cdot \mathbf{H}_{\mathcal{C}}^{\mathrm{T}} \neq 0$ which leads to $\hat{\boldsymbol{p}}_1^C \neq \hat{\boldsymbol{p}}_2^C$. □

Example 5.3. *Suppose the number of packets in every generation is* $n = 15$ *and each packet in the network contains linear combinations of at most* $m = 2$ *packets which leads to* $d = 2m + 1 = 5$. *Let also* $q = 2^4$. *The code* $\mathcal{C}$ *can be chosen to be the Reed-Solomon code with parameters* $\mathcal{C} = [15, 11, 5]_{2^4}$. *The parity check matrix of* $\mathcal{C}$ *can be written as follows*

$$\mathbf{H}_{\mathcal{C}} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{15-1} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(15-1)} \\ 1 & \alpha^3 & \alpha^6 & \cdots & \alpha^{3(15-1)} \\ 1 & \alpha^4 & \alpha^8 & \cdots & \alpha^{4(15-1)} \end{bmatrix}, \tag{5.7}$$

where α *is a primitive element of* $\mathbb{F}_{2^4}$. *Each column of* $\mathbf{H}_{\mathcal{C}}$ *can be assigned to one of* $n = 15$ *source packets.*

5.3.2 Decoding

Upon receiving a packet $\boldsymbol{p}$ with compressed coding vector $\hat{\boldsymbol{p}}^C$, the receiver needs to recover the original coding vector to construct the system of linear equation in (5.1).

In our construction, the problem of finding the original coding vector $\boldsymbol{p}^C$ from the compressed coding vector $\hat{\boldsymbol{p}}^C$ reduces to a decoding problem. In the coding theory terminology, we need to find the error vector having access only to the syndrome of a received vector. More formally, we may write

$$\begin{aligned} \text{find} \quad & \boldsymbol{p}^C \\ \text{subject to} \quad & \mathrm{hwt}(\boldsymbol{p}^C) \leq m, \\ & \boldsymbol{p}^C \cdot \mathbf{H}_{\mathcal{C}}^{\mathrm{T}} = \hat{\boldsymbol{p}}^C. \end{aligned} \tag{5.8}$$

This problem is in general NP-complete [49]. However, coding theory identifies instances that accept efficient encoding and decoding algorithms, and we leverage these constructions.

Note that, it is sufficient to find what are the non-zero positions of $\boldsymbol{p}^C$. If we know the non-zero positions, using the knowledge of the matrix $\mathbf{H}_{\mathcal{C}}$, we can uniquely recover the linear coefficients in the original coding vectors. The following lemma from coding theory formalizes this observation [50].

Lemma 5.2. *Let* $\mathcal{C}$ *be a linear code in* $\mathbb{F}_q^n$ *with parity check matrix* $\mathbf{H}_{\mathcal{C}}$. *Assume a codeword* $\boldsymbol{x}$ *is sent and a word* $\boldsymbol{y}$ *is received, with error vector* $\boldsymbol{e}$, *where* $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e}$. *Suppose we know a set* J *with at most* $d(\mathcal{C}) - 1$ *elements that*

Table 5.2 – Time for exhaustive search in seconds. Experiments are run on a single core of an Intel Centrino Duo2, at 3 GHz.

n/m	2	3	4
15	0.00018	0.0020	0.017
31	0.00097	0.024	0.48
63	0.0047	0.24	10.4

contains the set of error positions; i.e., non-zero elements of $\boldsymbol{e}$*. Then the error vector* $\boldsymbol{e}$ *is the unique solution of the following linear equations:*

$$\hat{\boldsymbol{e}}\mathbf{H}_{\mathcal{C}}^{\mathrm{T}} = \boldsymbol{y}\mathbf{H}_{\mathcal{C}}^{\mathrm{T}} \text{ and } \hat{e}_j = 0 \text{ for all } j \notin J. \tag{5.9}$$

The above lemma shows that we can reduce the problem of recovering the original coding vector to the problem of finding the non-zero positions of the original coding vector.

One approach to achieve this is through exhaustive search. For small values for m and n this in a fast computer can be feasible, as Table 5.2 illustrates. However, there are $\binom{n}{m}$ possible m-sets of non-zero positions to consider. This number grows exponentially in n when $\frac{m}{n}$ converges to a non-zero number.

A more practical approach is to use some known algebraic codes for $\mathcal{C}$ like BCH code, Reed-Solomon code [51], Goppa code [52], algebraic geometry codes [53], etc., to recover the original coding vectors efficiently. For all of the codes mentioned above there exists a version of the Berlekamp-Massey algorithm [54], [55] which allows the receivers to find the location of non-zero elements of original coding vectors as well as their values, using only the syndrome.

The Berlekamp-Massey algorithm consists of three stages that can be briefly summarized as follows. The first stage is the calculation of syndrome which in our approach we have it for free, since the received compressed coding vectors are equivalent to syndromes of original coding vectors. The second stage is to find the error locator polynomial which is defined as following

$$\lambda(z) \triangleq \prod_{r=1}^{\tau}(1 - \alpha^{i_r} z) = \sum_{r=0}^{\tau} \lambda_i z^r, \tag{5.10}$$

where $i_1, \ldots, i_\tau$ are the non-zero elements of $\boldsymbol{p}^C$, $\tau \leq t$, and α is a primitive nth root of unity. Finally, receivers find the roots of $\lambda(z)$ to find the location of non-zero components of $\boldsymbol{p}^C$ and using Lemma 5.2 can retrieve the original coding vectors.

5.3.3 Benefits

Using the compressed coding vectors method, the length of coding vectors reduces from n to $r = n - k$. The following lemma shows the optimality of our construction in terms of r.

Lemma 5.3. *The proposed construction leads to the shortest length possible for compressed coding vectors.*

Proof. The problem of finding the shortest representation for a sparse vector of length n and sparsity at most m over $\mathbb{F}_q$ is equivalent to the problem of designing the highest rate code with length n and minimum distance $2m+1$ over $\mathbb{F}_q$. Indeed, if one could find a smaller representation for the compresses coding vectors that still lets recovering the original coding vectors, i.e., optimization problem given by (5.8) is solvable, this implies that there exist a higher rate code with the specified parameters. □

We now examine what is the required size r for different cases. From the Singleton bound for code $\mathcal{C}$ we have

$$k \leq n - d + 1 = n - \min[2m, n],$$

so for $\frac{n}{2} \leq m \leq n$ we have $k = 0$ which implies that we can select w.l.o.g. the full rank $n \times n$ parity matrix $\mathbf{H}_{\mathcal{C}}$ to be the identity matrix. In this case, we recover the usual NC with coding vectors $\{\boldsymbol{e}_i\}_{i=1}^{n}$ appended to the sources packets, and there is no benefit from our approach.

From the Gilbert-Varshamov bound [56] we have an upper bound for the length of compressed coding vectors r that for the case $m < \frac{n}{4}$ can be simplified to

$$r \leq nH_q\left(\frac{d-1}{n}\right) = nH_q\left(\frac{2m}{n}\right), \tag{5.11}$$

where $H_q(\delta) = \delta \log_q(q-1) - \delta \log_q \delta - (1-\delta)\log_q(1-\delta)$ is the q-ary entropy function. Also, the Sphere packing bound leads to a lower bound on the length of compressed coding vectors where for $m < \frac{n}{2}$ we can simplify it to obtain

$$\begin{aligned} r &\geq nH_q\left(\frac{d-1}{2n}\right) - \frac{1}{2}\log_q\left(4(d-1)\left(1 - \frac{d-1}{2n}\right)\right) \\ &= nH_q\left(\frac{m}{n}\right) - \frac{1}{2}\log_q\left(8m\left(1 - \frac{m}{n}\right)\right). \end{aligned} \tag{5.12}$$

From (5.11) and (5.12), for fixed values of m, and as the number of source packets grows, we have

$$m \log_q n + O(1) \leq r \leq 2m \log_q n + O(1),$$

So using the proposed method, we can reduce the growth of coding vectors from $O(n)$ to $O(m \log n)$.

Example 5.4. *Using a table of the best codes known (from [48] and [57]), we can see for example that, there exist binary linear codes of length* $n = 127$ *with redundancy* $r = 35$ *and minimum distance* $d = 2m + 1 = 11$*, which is in fact a shortened version of* $[128, 93, 11]_2$ *Goppa code [52]. Thus in a network with* 127 *source packets in each generation if at most* $m = 5$ *source vectors get combined, we need to use coding vectors of length* $r = 35$ *instead of* $n = 127$*.*

In the previous example, it is assumed that the network nodes perform binary NC; i.e., nodes only XOR the packets. However, if the field size is increased, a shorter compressed coding vectors can be used as it is shown in the

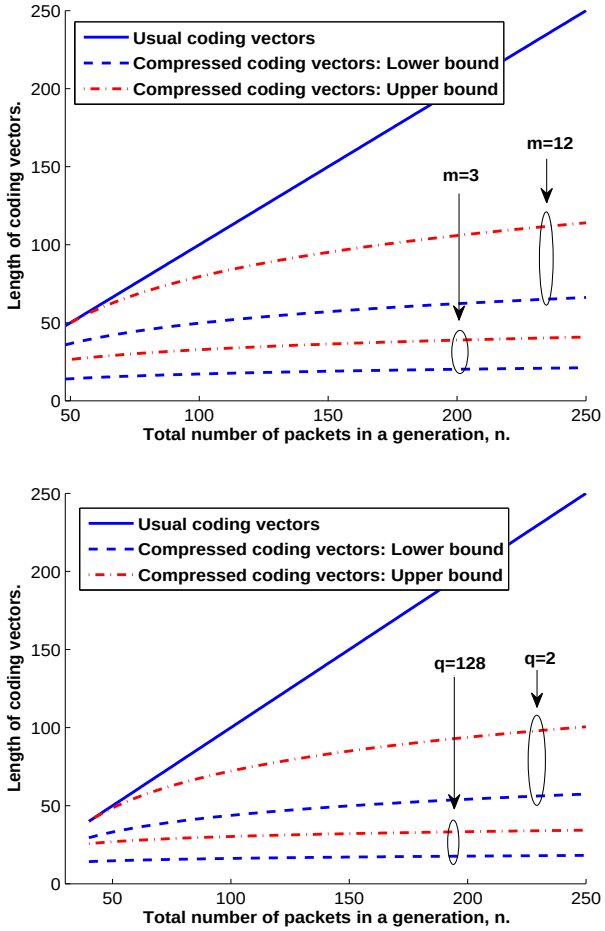


Figure 5.1 – Bounds on the length of the compressed coding vectors, r. Upper figure: r as a function of the number of packets in a generation n, when $m = 2$ and $m = 20$ sources get combined, $q = 2$. Lower figure: r as a function of n when $m = 10$ sources get combined, for two values of field size $q = 2$ and $q = 128$.

Example 5.5. In fact the code used in Example 5.4 is not a MDS [3] code while the Reed-Solomon code in Example 5.5 is MDS.

Example 5.5. *It is known that the Reed-Solomon code [51] is a linear code* $[n, k, d]_q$ *where* $n = q - 1$ *and* $k + d - 1 = n$. *To compare the length of compressed coding vectors resulting from the Reed-Solomon code as in Example 5.4, consider a field of size* $q = 2^7$, *which leads to* $n = 127$. *Also set*

3. Maximum distance separable.

$d = 2m + 1 = 11$ *for the redundancy; the length of compressed coding vectors equals* $r = n - k = d - 1 = 10$ *where the ratio of the compressed coding vector length to the original coding vector length is much lower than that of Example 5.4. Compared to the case of classical coding vectors, the coding vector headers decrease from* 112 *Bytes to only* 9 *Bytes.*

5.3.4 Effect on Rate

A natural question to ask is, if we restrict the number of combined packets, how does this affect the observed multicasting rate. This clearly depends not only on the value of m, but also on the network topology and on the subsets of m packets that get combined. For example, for some networks, no coding is required to achieve the min-cut rate for all receivers. It is also easy to come up with specifically constructed examples, where we cannot achieve the min-cut rate unless all source packets get combined.

However, as we argued in Section 5.1, in many situations, such as the case of multi-source wireless networks, it is not practical to allow all possible linear combinations to occur. Additionally, in preliminary experiments we are performing, we see that the number of actually combined packets depends on the distance from the receiver and the number of sources in the vicinity, and can be much smaller than the total number of sources in the network, as sources that are topologically separated may very rarely have their packets combined.

One possible way to abstract this problem is the following. Consider again the linear equations that the receiver needs to solve in (5.1) and assume that the non-zero elements in the $n \times n$ matrix $\boldsymbol{H}$ are chosen uniformly at random with probability m/n. For each non-zero position, a uniformly at random non-zero coefficient from the field $\mathbb{F}_q$ is then selected. This will result in a sparse matrix $\boldsymbol{H}$, where each row will have on the average m non-zero elements from $\mathbb{F}_q$ per row (coding vector). From [58], we have the following lemma.

Lemma 5.4. *For every* $c \geq 0$ *there exist a constant* a_c *such that for the random matrix* $\boldsymbol{H} \in \mathbb{F}_q^{n \times n}$, $n > e^c$, *with* $m = \log n - c$ *we have*

$$\mathbb{P}\left[\operatorname{rank}(\boldsymbol{H}) < n\right] \leq \frac{a_c}{q}. \tag{5.13}$$

Proof. See [58, Corollary 2.4]. □

The work in [59] has extended the above lemma and showed that for $m > \log n$ the probability that the matrix $\boldsymbol{H}$ is not full rank approaches zero polynomially fast with n. The above argument shows that m should be at least order $O(\log n)$ to let receivers end up with full rank matrix $\boldsymbol{H}$ with high probability.

5.4 Concluding Remarks

In this chapter, we have presented a novel approach for practical NC that uses shortened coding vectors as compared to the classical approach. We showed

that we can reduce the length of the coding vectors from n to $O(m \log n)$ where n is the number of source packets injected in the network, if each coded packet contains the combination of at most m source packets.

5.4.1 Joint Identity-Message Coding

We may combine ideas from this chapter and Chapter 3 (Section 3.4) to propose a coding scheme for sensor network applications where the identity of the sensor nodes form the bulk of the messages being transmitted towards a destination. To this end, we have proposed a communication protocol for such networks, where sensor identities and measurements are jointly encoded in fixed-size vectors. We refer interested readers to [60, 23].

The basic idea behind the proposed protocol is to assign a different codebook to each sensor and let each sensor implicitly convey its identity through its choice of codebook. We realize this using subspace encoding: each reporting sensor communicates its identity by generating a set of vectors that represent a distinct subspace—distinct from the subspaces generated by all other sensors. By combining incoming vectors, intermediate nodes essentially produce different (compact) representations of the subspaces generated by the reporting sensors.

The benefits of this scheme consist of balancing the transmission load across all nodes in the network (important for networks that are periodically recharged through natural resources), low-complexity network operations, and graceful incorporation of error resilience.

Part II

Subspace Properties of Network Coding

"In theory, theory and practice are the same. In practice, they are not."

- Albert Einstein

6 Subspace Properties of Network Coding and their Applications

Randomized NC offers a promising technique for content distribution systems. In randomized NC, each node in the network combines its incoming packets randomly and sends them to its neighbors [4, 20]. This is the approach adopted by most practical applications today. For example, Avalanche, the first implementation of a peer-to-peer (P2P) system that uses NC, adopts such a randomized operation [61, 62]. In ad-hoc wireless and sensor networks as well, most proposed protocols employing NC again opt for randomized network operation (see [63] and references therein).

Our contributions start with the observation that coding vectors (more generally message packets) implicitly carry information about the network structure as well as its state [1]. Such vectors belong to appropriately defined vector spaces, and we are interested in fundamental properties of these vector spaces (defined over a finite field). In particular, since we are investigating properties induced by randomized NC, we need to characterize random subspaces of the aforementioned vector spaces. These properties of random subspaces over finite fields might be of independent interest. We aim to show, using these properties, that observing the coding vectors we can passively collect structural and state information about a network. We can leverage this information towards several applications that are interesting in their own merit, such as topology inference, network tomography, and network management (we do not claim here the design of practical protocols that use these properties). However, we show that randomized NC, apart from its better known properties for facilitating information delivery, can provide us with information about the network itself.

To support this claim, we start by studying the problem of passive topology inference in a content distribution system where intermediate nodes perform randomized NC. We show that the subspaces nodes collect during the dissem-

1. By state we refer to link or node failures, congestion in some part of the network, etc.

ination process have a dependence with each other which is inherited from the network structure. Using this dependence, we describe the conditions that let us perfectly reconstruct the topology of a network, if subspaces of all nodes at some time instant are available.

We then investigate a reverse or dual problem of topology inference, which is, finding the location of Byzantine attackers. In a network coded system, the adversarial nodes in the network can disrupt the normal operation of information flow by inserting erroneous packets into the network. We use the dependence between subspaces gathered by network nodes and the topology of the network to extract information about the location of attackers. We propose several methods, compare them and investigate the conditions that allow us to find the location of attackers up to a small uncertainty.

Finally, we observe that the received subspaces, even at one specific node, reveal some information about the network, such as the existence of bottlenecks or congestion. We consider P2P networks for content distribution that use randomized NC techniques. It is known that the performance of such P2P networks depends critically on the good connectivity of the overlay topology. Building on our observation, we propose algorithms for topology management to avoid bottlenecks and clustering in network-coded P2P systems. The proposed approach is decentralized, inherently adapts to the network topology, and reduces substantially the number of topology rewirings that are necessary to maintain a well connected overlay; moreover, it is integrated in the normal content distribution.

6.1 Related Work

Network coding started by the work of Ahlswede et al. [6] who showed that a source can multicast information at a rate approaching the smallest min-cut between the source and any receiver if the middle nodes in the network combine the information packets. Li et al. [5] showed that linear NC with finite field size is sufficient for multicast. Koetter et al. [7] presented an algebraic framework for linear NC.

Randomized NC was proposed by Ho et al. [4, 64] where they showed that randomly choosing the network code leads to a valid solution for a multicast problem with high probability if the field size is large. It was later applied by Chou et al. [20] to demonstrate the practical aspects of random linear NC. Gkantsidis et al. [61, 62] implemented a practical file sharing system based on this idea. Several other works have also adopted randomized NC for content distribution, see for example [65, 66], and [67].

Network error correcting codes, that are capable of correcting errors inserted in the network, have been developed during the last few years. For example see the work of Koetter et al. [1], Jaggi et al. [68], Ho et al. [69], Yeung et al. [70, 71], Zhang [72], and Silva et al. [73]. These schemes are capable of delivering information despite the presence of Byzantine attacks in the network or nodes malfunction, as long as the amount of undesired information is limited. These

network error correcting schemes allow to correct malicious packet corruption up to certain rate. In contrast, we use NC to identify malicious nodes in our work. Recently, and following our work [74], additional approaches are proposed in the literature, some building on our results [75].

Overlay topology monitoring and management that do not employ network coding has been an intensively studied research topic, see for example [76]. However, in the context of NC, it is a new area of research. Fragouli et al. [77, 78] took advantage of NC capabilities for active link loss network monitoring where the focus was on link loss rate inference. Passive inference of link loss rates has also been proposed by Ho et al. [79]. In a subsequent work of ours, Sharma et al. [37] study passive topology estimation for the upstream nodes of every network node. This work is based on the assumption that the local coding vectors for each node in the network are fixed, generated in advance and known by all other nodes in the network, unlike our work that builds on randomized operation. The idea of passive inference of topological properties from subspaces that are build over time, as far as we know, is a novel contribution of this work.

6.2 Models: Coding and Network Operation

A simple observation motivates much of the work presented in this chapter: the subspaces gathered by the network nodes during information dissemination with randomized NC, are not completely random, but have some relationship, and this relationship conveys information about the network topology as well as its state. We will thus investigate properties of the collected subspaces and show how we can use them for diverse applications. To this end, we will use the basic results developed in Section 2.4 for the randomly sampled subspaces over a finite field.

Different properties of the subspaces are relevant to each particular application and therefore we will develop a framework for investigating these properties. This will also involve some understanding of modeling the problem to fit the requirements of an application and then developing subspace properties relevant to that model.

6.2.1 Notation

In this chapter, we are interested in investigating the relationship of the collected subspaces at neighboring network nodes. We consider a network represented as a directed acyclic graph $G = (V, E)$, with $\vartheta = |V|$ nodes and $\xi = |E|$ edges. If a node u has p parents $u_1, \ldots, u_p$, we denote with $P(u) = \{u_1, \ldots, u_p\}$ the set of parents of u. We use $P^l(u)$ to denote the set of all ancestors of u at distance l from u in the network (we say that two nodes u and v are at distance l if there exists a path of length exactly l that connects them). We denote with $\pi_u^{(u_i)}(t)$ the subspace node u receives from parent u_i at exactly time t, and with $\pi_u(t)$ the whole subspace (from all parents) that node u receives at time t, that is $\pi_u(t) = \sum_{i=1}^{p} \pi_u^{(u_i)}(t)$. We also denote with

$\Pi_u^{(u_i)}(t)$ the subspace node u has received from parent u_i up to time t, that is, $\Pi_u^{(u_i)}(t) = \Pi_u^{(u_i)}(t-1) + \pi_u^{(u_i)}(t)$. Then the subspace $\Pi_u(t)$ that the node has at time t can be expressed as $\Pi_u(t) = \sum_{i=1}^{p} \Pi_u^{(u_i)}(t)$. For a set of nodes $\mathcal{U} = \{u_1, \ldots, u_p\}$, we define $\Pi_{\mathcal{U}} = \Pi_{u_1} + \cdots + \Pi_{u_p}$.

6.2.2 Network Operation

We assume that there is an information source located on a node S that has a set of n packets (messages) $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, $\boldsymbol{x}_i \in \mathbb{F}_q^L$, to distribute to a set of receivers, where each packet is a sequence of L symbols over the finite field $\mathbb{F}_q$. To do so, we will employ a dissemination protocol based on randomized network coding, namely, where each network node sends random linear combinations (chosen to be uniform over $\mathbb{F}_q$) of its collected packets to its neighbors. We assume for simplicity that there are no packet-losses.

6.2.2.1 Dissemination Protocol

It is possible to separate the dissemination protocols into the following operation categories.

- *Synchronous:* All nodes are synchronized and transmit to their neighbors according to a global clock tick (time-slot). At time-slot $t \in \mathbb{N}$, node v sends linear combinations from all vectors it has collected up to time $t-1$. Once nodes start transmitting information, they keep transmitting until all receivers are able to decode.
- *Asynchronous:* Nodes transmit linear combinations at randomly and independently chosen time instants.

In this chapter, we focus on the synchronous network where we assume that each link has unit delay [2] corresponding to each time-slot, however our results can be extended to asynchronous networks as well.

Next, we explain in detail the dissemination protocol, that is summarized in Algorithm 6.1.

Timing: We depict in Figure 6.1 the relative timing of events within a time-slot. Nodes transmit at the beginning of a time-slot. We assume that each packet is received by its intended receiver before the end of the time-slot. Thus, the time-slot duration incorporates the packet propagation delay in one edge of the network.

Rate Allocation and Equivalent Network Graph: The dissemination protocol first associates with each link e of the network a rate r_e (measured as the number of packets transmitted per time-slot on edge e). These rates are selected in advance using a rate allocation method, for example see [80].

2. Unit delay can model a buffering window a node needs to wait to collect packets from all its neighbors.

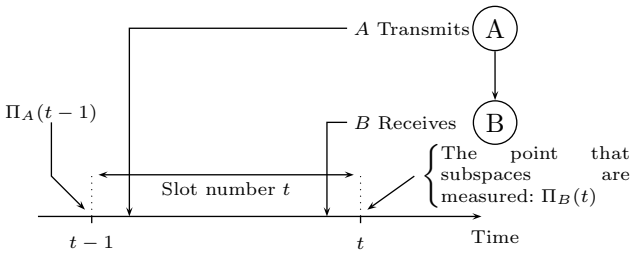


Figure 6.1 – Timing schedule of the dissemination protocol given by Algorithm 6.1.

For the rest of the chapter, we consider an equivalent network graph, where each edge e has capacity equal to its allocated rate r_e. On this new graph, we can define the min-cut c_v from the source node S to a node $v \in V$. Whenever we refer to min-cut values in the following, we imply min-cut values over this equivalent graph.

We assume that the rate allocation protocol we use satisfies

$$r_e \leq \min[c_e, c_{\text{tail}(e)}], \tag{6.1}$$

where c_e is the capacity of edge e. This very mild assumption says that the node $v = \text{tail}(e)$ does not send more information than it receives, and is satisfied by all protocols that do not send redundant packets.

In this chapter, we consider the case where $n \gg c_v$, namely, the dissemination of the n source packets to the receivers takes place by using the network over several time-slots.

Node Operation: When the dissemination starts, at time-slot say zero, the source starts transmitting at each time-slot and to each of its outgoing edges e, r_e randomly selected linear combinations of n information packets. We will call r_S the *source rate*. The source continues until it has transmitted linear combinations of all n packets, i.e., for $\frac{n}{r_S}$ times-lots. Every other node $v \in V \setminus \{S\}$ in the network, operates as follows:

- Initially it does not transmit, but only collects in a buffer packets from its parents, until a time τ_v, which we call *waiting time* and we will define in the following. As we will see, each node can decide the waiting time by itself and independently from other nodes.
- At each time-slot t, for all $t \geq \tau_v + 1$, it transmits to each outgoing edge e, r_e linear combinations of all packets it has collected in its buffer up to time $t-1$.

Collected Subspaces: We can think of each of the n source messages $\{\boldsymbol{x}_i\}_{i=1}^n$ as corresponding to one dimension of an n-dimensional space $\Pi_S \sqsubseteq \mathbb{F}_q^L$ where $\Pi_S = \langle \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \rangle$. We say that node $v \in V$ at time t observes a subspace $\Pi_v(t) \sqsubseteq \Pi_S$, with dimension $d_v(t) \triangleq \dim(\Pi_v(t))$, if $\Pi_v(t)$ is the space spanned

by the received vectors at node v up to time t. Initially, at time $t = 0$, the collected subspaces of all nodes (apart the source) are empty; $d_v(0) = 0$, $\forall v \in V \setminus \{S\}$.

Waiting Times: We next define the waiting times, that will be used in the following sections to ensure that the subspaces of different nodes be distinct, and are a usual assumption in dissemination protocols; indeed, for large n the waiting time does not affect the rate. For example, in the information-theoretic proof of the main theorem in NC [6], each node waits until it collects at least one message from each of its incoming links before starting transmissions.

Definition 6.1. *The* waiting time τ_v *for a node v is the first time-slot during which node v receives information from the source at a rate equal to its min-cut c_v, and additionally, has collected in its buffer a subspace of dimension at least $c_v + 1$.*

Note that, because we are dealing with acyclic graphs, we can impose a partial order on the waiting times of the nodes, such that all parents of a node have smaller waiting time than the node. Moreover, each node can decide whether the conditions for the waiting time are met, by observing whether it receives information at a rate equal to its min-cut, and what is the dimension of the subspace it has collected. That is, a node does not need to know any topological information (apart from its min-cut), and the waiting times do not need to be communicated in advance to the nodes, but can be decided online based on the network conditions.

Algorithm 6.1 Dissemination protocol.

Input: $G(V, E)$, S, n, $\{\tau_v\}$, $\{r_e\}$
1: **for all** $v \in V \setminus \{S\}$ **do**
2: $\Pi_v(0) = \varnothing$, $d_v(0) = 0$;
3: **end for**
4: $t \leftarrow 1$
5: **while** $\min_v d_v(t) < n$ **do**
6: **for all** $v \in V$ **do**
7: **if** $t \geq \tau_v + 1$ **then**
8: **for all** $e \in \mathrm{Out}(v)$ **do**
9: node v transmits from $\Pi_v(t-1)$ on e with rate r_e;
10: **end for**
11: **end if**
12: **end for**
13: **for all** $v \in V$ **do**
14: update $\Pi_v(t)$ and $d_v(t)$;
15: **end for**
16: $t \leftarrow t + 1$;
17: **end while**

6.2.2.2 Source Operation and the Source Subspace Π_S

As we discussed, the source needs to convey to the receivers n source packets that span the n-dimensional subspace $\Pi_S = \langle \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \rangle$, with $\Pi_S \sqsubseteq \mathbb{F}_q^L$. Π_S is isomorphic to $\mathbb{F}_q^n$; thus, for the purpose of studying relationships between subspaces of Π_S, we can equivalently assume that $\Pi_S = \mathbb{F}_q^n$, and that node $v \in V$ at time t observes a subspace $\Pi_v(t) \sqsubseteq \Pi_S$. This simplification is very natural in the case where we employ coding vectors, reviewed briefly in the following (see also Section 2.2), as we only need consider the coding vectors for our purposes and ignore the remaining contents of the packets; however, we can also use the same approach in the case where the source performs non-coherent coding, described subsequently.

Use of Coding Vectors: To enable receivers to decode, as we have seen in Section 2.2, the source assigns n symbols of each message packet to determine the linear relation between that packet and the original packets $\boldsymbol{x}_i$, $i = 1, \ldots, n$. Each message vector $\boldsymbol{x}_i$ contains two parts. The vector $\boldsymbol{x}_i^C \in \mathbb{F}_q^n$ with length n is the coding vector and remaining part, $\boldsymbol{x}_i^I \in \mathbb{F}_q^{L-n}$, is the information part where

$$\boldsymbol{x}_i = [\boldsymbol{x}_i^C \mid \boldsymbol{x}_i^I]. \tag{6.2}$$

For our purposes, it is sufficient to restrict our algorithms to examine the coding vectors. Thus, the source has the space $\Pi_S = \mathbb{F}_q^n$; during the information dissemination, if a node v at time t has collected m packets $\boldsymbol{y}_i$ with coding vectors $\boldsymbol{y}_i^C$, it has observed the subspace $\Pi_v(t) = \langle \boldsymbol{y}_1^C, \ldots, \boldsymbol{y}_m^C \rangle$. In other words, the coding vectors capture all the information we need for our applications.

Subspace Coding: Our approach also works in the case of subspace coding [1], that was briefly explained in Section 2.2.

In the case of subspace coding, the dissemination algorithm works in exactly the same way as in the case of coding vectors; what changes is how the source maps the information to the packets it transmits, and how decoding occurs. However, this is orthogonal to our purposes, since we perform no decoding of the information messages, but simply observe the relationship between the subspaces different nodes in the network collect. Thus, the same approach can be applied in this case as well.

6.2.3 Input to Algorithms

We are interested in designing algorithms that leverage the relationships between subspaces observed at different network nodes for network management and control. The algorithms design will depend on the information that we have access to. We distinguish between the following.

- *Global information*: A central entity knows the subspaces that all ϑ nodes in the network have observed.
- *Local Information:* There is no such omniscient entity, and each node v only knows what it has received, its own subspace Π_v.

We may also have information between these two extreme cases. Moreover, we may have a *static view*, where we take a snapshot of the network at a given time instant t, or a *non-static view*, where we take several snapshots of the network and use the subspaces' evolution to design an algorithm.

We will argue in Section 6.4 that capturing even global information can be accomplished with relatively low overhead (sending one additional packet per node at the end of the dissemination protocol); thus, the algorithms we develop even assuming global information can in fact be implemented almost passively and at low cost.

6.3 Rate of Innovative Packets

In the following sections, we will need to know the rate of receiving innovative message vectors (packets) at receivers in a dissemination protocol performing randomized NC. By innovative we refer to vectors that do not belong in the space spanned by already collected packets. As it is shown in [6], the source can multicast at rate equal to the minimum min-cut of all receivers if the intermediate nodes can combine the incoming messages. Moreover, it is shown in [5] that using linear combinations is sufficient to achieve information transfer at a rate equal to the minimum min-cut of all receivers. In [6, 4], it is also demonstrated that choosing the coefficients of the linear combinations randomly is sufficient (no network-specific code design is required) with high probability if the field size is large enough.

To find the rate of receiving information at each node where the implemented dissemination protocol performs randomized network coding, we can use the following result given in Theorem 6.1. Note that our described dissemination protocol, although very common in practice, does not exactly fit to the previous theoretical results in the literature that examine rates, because the operation of the network nodes is not memory-less. That is, while for example in [4, 6, 5] each transmitted packet at time t is a function of a small subset of the received packets up to time t (the ones corresponding to the same information message), in our case a packet transmitted at time t is a random linear combination of all packets received up to time t. This small variant of the main theorem on randomized NC is very intuitive, and we formally state it in following.

Theorem 6.1. *Consider a source that transmits n packets over a connected network using the dissemination protocol described in Section 6.2.2, and assume that the network nodes perform random linear NC over a sufficiently large finite field. Then there exists t_0 such that for all $t > t_0$ each node v in the network receives c_v independent linear combinations of the n source packets per time slot, where $c_v = \text{min-cut}(v)$.*

Proof. Refer to Appendix 6.C. □

Given Theorem 6.1, we can state the following definition.

Definition 6.2. *For a specific information dissemination protocol over a network, we define the* steady state *as the time period during which each node v in the network receives exactly c_v independent linear combinations of the n source packets per time slot and none of the nodes, except source S, has collected n linearly independent combinations. We call the time that the network enters steady state phase the* steady state starting time *and denote it by T_s. If the network never attains the steady state phase then we set $T_s = \infty$.*

For our protocol in Section 6.2.2, T_s depends not only on the network topology, but also on the waiting times τ_v. For the waiting time defined in Definition 6.1 we can upper bound T_s as stated in Lemma 6.1.

Lemma 6.1. *If n is large enough, for the dissemination protocol given in Section 6.2.2 we may upper bound the steady state starting time as follows*

$$T_s \leq 2D(G), \tag{6.3}$$

where $D(G)$ is the longest path from the source to other nodes in the network[3]*.*

Proof. Refer to Appendix 6.A. □

In order to be sure that the dissemination protocol given in Section 6.2.2 enters the steady state phase, n should be large enough. Using Lemma 6.1 we have the following result, Corollary 6.1.

Corollary 6.1. *A sufficient condition for n to be sure that the protocol enters the steady state is that*

$$2D(G) < \lfloor \frac{n}{c_{max}} \rfloor, \tag{6.4}$$

where $c_{max} = \max_{v \in V} c_v$.

6.4 Topology Inference

In this section, we will use the tools developed in Section 2.4 to investigate the relation between the network topology and the subspaces collected at the nodes during information dissemination. We will develop conditions that allow us to passively infer the network topology with (asymptotically on the value of q) no error. The proposed scheme is passive in the sense that it does not alter the normal data flow of the network, and the information rates that can be achieved. In fact, we can think of our protocol as identifying the topology of the network which is induced by the traffic.

We build our intuition starting from information dissemination in tree topologies, and then extend our results in arbitrary topologies. Note that information dissemination using NC in tree topologies does not offer throughput benefits as compared to routing; however, it is an interesting case study that

3. Note that $D(G)$ is different from the longest shortest path which is called diameter of G in the graph theory literature.

will naturally lead to our framework for general topologies. We then define conditions under which the topology of a tree and that of an arbitrary network can be uniquely identified using the observed subspaces. Note that uniquely identifying the topology is a strong requirement, as the number of topologies for a given number of network nodes is exponential in the number of nodes.

6.4.1 Tree Topologies

Let $G = (V, E)$ be a network that is a directed tree of depth $D(G)$, rooted at the source node S. We will present (i) necessary and sufficient conditions under which the tree topology can be uniquely identified, and (ii) given that these conditions are satisfied, algorithms that allow us to do so.

We first consider trees where each edge is allocated the same rate c, and thus the min-cut from the source to each node of the tree equals c. We then briefly discuss the case of undirected trees. Finally we examine the case where edges are allocated different rates, and thus nodes may have different min-cuts from the source.

6.4.1.1 Common Min-Cut

Assume that each edge of the tree has the same capacity c (i.e., a rate allocation algorithm has assigned the same rate $r_e = c$ on each edge of the tree). Thus all nodes in the tree have the same min-cut, equal to c. Then according to the dissemination protocol introduced in Algorithm 6.1, each node v will wait time τ_v, until it has collected a $c + 1$ dimensional subspace, and then start transmitting to its children. Our claim is that, we can then identify the network topology using a single snapshot of all nodes' subspaces at a time t. Before formally proving the result in Theorem 6.2, we will give some intuition on why this is so, and why the waiting time is crucial to achieve this. We start from an example on the simple network in Figure 6.2.

Example 6.1. *Consider the tree in Figure 6.2 and assume that the edges have unit capacity ($c = 1$). Algorithm 6.1 works as follows. At time $t = 1$, node A receives a vector $\boldsymbol{y}_1$ from the source S. Node A waits, as it has not yet collected a $(c + 1) = 2$ dimensional subspace. At time $t = 2$, it receives a vector $\boldsymbol{y}_2$. It now has collected the subspace $\Pi_A(2) = \langle \boldsymbol{y}_1, \boldsymbol{y}_2 \rangle$, and thus at the next time-slot it will start transmitting. At time $t = 3$, node A transmits vectors $\boldsymbol{y}_1^B$ and $\boldsymbol{y}_1^C$ to nodes B and C respectively, with $\boldsymbol{y}_1^B, \boldsymbol{y}_1^C \in \Pi_A(2)$. Thus $\Pi_B(3) = \langle \boldsymbol{y}_1^B \rangle$ and $\Pi_C(3) = \langle \boldsymbol{y}_1^C \rangle$. Node A also receives a vector $\boldsymbol{y}_3$ from the source, and thus $\Pi_A(3) = \langle \boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3 \rangle$. Consider now the subspaces $\Pi_A(3)$, $\Pi_B(3)$ and $\Pi_C(3)$. We see that $\Pi_B(3) \sqsubseteq \Pi_A(3)$, and $\Pi_C(3) \sqsubseteq \Pi_A(3)$; we thus conclude that nodes B and C are children of node A. Moreover, $\Pi_B(3) \neq \Pi_C(3)$, which will allow us to distinguish between children of these two nodes when we deal with larger trees.*

In contrast, if Algorithm 6.1 did not impose a waiting time, and node A started transmitting to nodes B and C at time $t = 2$, then both nodes B and C

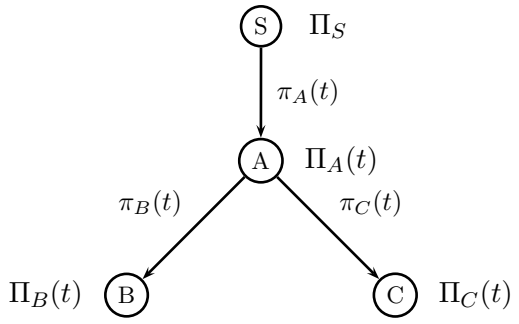


Figure 6.2 – Directed tree with four nodes rooted at the source S.

would receive the same subspace $\langle \boldsymbol{y}_1 \rangle$*, i.e.,* $\Pi_B(2) = \Pi_C(2) = \langle \boldsymbol{y}_1 \rangle$*. In fact, at all subsequent times, we will have that* $\Pi_B(t) = \Pi_C(t) = \Pi_A(t-1)$*. Thus, we would not be able to distinguish between these two nodes.*

The main idea in our result is that, if we consider two nodes u and v in the network which have collected subspaces $\Pi_u(t)$ and $\Pi_v(t)$ at time t, then, unless u and v have a child-ancestor relationship (i.e., are on the same branch in the tree), it holds that $\Pi_u(t) \nsubseteq \Pi_v(t)$ and $\Pi_v(t) \nsubseteq \Pi_u(t)$.

The challenge in proving this is that we deal with subspaces evolving over time, and thus we cannot directly apply the results in Section 2.4. For example, for the network in Figure 6.2, $\Pi_B(t)$ and $\Pi_C(t)$ are not subspaces that are selected uniformly at random from $\Pi_A(t)$; instead, they are build over time as $\Pi_A(t)$ also evolves. We will thus need the following two results, that modify the results in Section 2.4 to take into account the time evolution in the creation of the subspaces. We start by examining in Lemma 6.2 the relationship between subspaces collected at the immediate children of a given parent node (for example, at the children B and C of node A). These are created by sampling the same subspaces (those at node A). We then examine in Corollary 6.2 the relationship between subspaces collected at nodes that have different parents (for example, a node that has B as parent and a node that has C as parent).

Lemma 6.2. *Suppose there exist (proper) subspaces* $\Pi(0) \sqsubset \Pi(1) \sqsubset \cdots \sqsubset \Pi(t-1)$ *with dimensions* $d_0, \ldots, d_{t-1}$*, respectively. Let us construct the set of subspaces* $\Pi_u(i)$*,* $i = 1, \ldots, t$*, as follows. Set* $\Pi_u(i) = \sum_{j=1}^{i} \pi_u(j)$ *where* $\pi_u(j)$ *is the span of* $k_u(j)$ *vectors chosen uniformly at random from* $\Pi(j-1)$ *such that* $k_u(1) < d_0$ *and* $k_u(j) \leq (d_{j-1} - d_{j-2})$ *for* $j = 2, \ldots, t$*. Similarly, we construct the set of subspaces* $\Pi_v(i) = \sum_{j=1}^{i} \pi_v(j)$ *where for* $k_v(j)$ *we have similar conditions, namely,* $k_v(1) < d_0$ *and* $k_v(j) \leq (d_{j-1} - d_{j-2})$ *for* $j = 2, \ldots, t$*. Then we have*

$$\Pi_u(i) \nsubseteq \Pi_v(j) \quad \textit{and} \quad \Pi_v(j) \nsubseteq \Pi_u(i) \quad \forall i, j \in \{1, \ldots, t\}, \tag{6.5}$$

with high probability.

Proof. Refer to Appendix 6.A. □

Corollary 6.2. *Suppose that there exist two set of subspaces* $\{\Pi_u(i)\}_{i=0}^{t-1}$ *and* $\{\Pi_v(i)\}_{i=0}^{t-1}$ *such that* $\Pi_u(0) \sqsubset \cdots \sqsubset \Pi_u(t-1)$ *and* $\Pi_v(0) \sqsubset \cdots \sqsubset \Pi_v(t-1)$. *Moreover, assume that* $\Pi_u(i) \not\subseteq \Pi_v(j)$ *and* $\Pi_v(j) \not\subseteq \Pi_u(i)$ $\forall i, j \in [0 : t-1]$. *Now, construct two set of subspaces* $\{\Pi_a(i)\}_{i=1}^{t}$ *and* $\{\Pi_b(i)\}_{i=1}^{t}$ *by setting* $\Pi_a(i) = \sum_{j=1}^{i} \pi_a(j)$ *and* $\Pi_b(i) = \sum_{j=1}^{i} \pi_b(j)$ *where* $\pi_a(i)$ *is chosen uniformly at random from* $\Pi_u(i-1)$ *and* $\pi_b(i)$ *is chosen uniformly at random from* $\Pi_v(i-1)$ *(with some arbitrary dimension). Then we have*

$$\Pi_a(i) \not\subseteq \Pi_b(j) \quad \textit{and} \quad \Pi_b(j) \not\subseteq \Pi_a(i) \quad \forall i, j \in \{1, \ldots, t\}, \tag{6.6}$$

with high probability.

Proof. Refer to Appendix 6.A. □

Theorem 6.2. *Consider a tree of depth* $D(G)$ *where each edge has capacity* c*, and the dissemination Algorithm 6.1. A static global view of the network at time* t*, with* $2D(G) < t < \lfloor \frac{n}{c} \rfloor$*, allows to uniquely determine the tree structure with high probability, if the waiting times are chosen according to Definition 6.1.*

Proof. We will say that a node of the tree is at level l if it has distance l from the source. In a tree there exists a unique path $\mathcal{P}_u = \{S, P^{l_u-1}(u), \ldots, P(u), u\}$ from source S to node u at level l_u of the network.

If we consider a time t in steady state (where all nodes have nonempty subspaces and none has collected the whole space), then clearly using Algorithm 6.1 for dissemination in the network for the nodes along the path $\mathcal{P}_u$ it holds that

$$\Pi_u(t) \sqsubset \Pi_{P(u)}(t) \sqsubset \cdots \sqsubset \Pi_{P^{l_u-1}(u)}(t) \sqsubset \Pi_S. \tag{6.7}$$

Note that the conditions on t ensure that the network is in steady-state.

To identify the topology of the tree it is sufficient to show that $\Pi_u(t) \not\subseteq \Pi_v(t)$ for any node v that is not in $\mathcal{P}_u$. Let l_u and l_v be the distance of u and v from the source, respectively.

First, we observe that, starting from the source, by applying Lemma 6.2 and Corollary 6.2 and because of Definition 6.1 the subspaces of the nodes at the same level (same distance from the source) are different at all times. So it only remains to check the condition $\Pi_u(t) \not\subseteq \Pi_v(t)$ for those node v that are not in the same level as u.

Consider two cases. First, if $l_u < l_v$ then let v' be the ancestor of v at the same level as u. By Corollary 6.2 we have $\Pi_u(t) \not\subseteq \Pi_{v'}(t)$ so $\Pi_u(t) \not\subseteq \Pi_v(t)$ because $\Pi_v(t) \sqsubseteq \Pi_{v'}(t)$.

Now consider the second case, $l_u > l_v$. We start by assuming $\Pi_u(t) \sqsubseteq \Pi_v(t)$ and then we will show that this assumption leads to a contradiction. Let u' be the ancestor of u at the same level of v. Then we make the following observation. If at time t we have $\Pi_u(t) \sqsubseteq \Pi_v(t)$ by Lemma 2.9 we should have

had $\Pi_{P(u)}(t-1) \sqsubseteq \Pi_v(t)$ and so $\Pi_{P^2(u)}(t-2) \sqsubseteq \Pi_v(t)$ and finally we should had had $\Pi_{u'}(t - l_u + l_v) \sqsubseteq \Pi_v(t)$. But according to Corollary 6.2 this is a contradiction because u' and v are at the same level.

In the above argument, we have shown that $\Pi_{P(u)}(t)$ is the smallest subspace contains $\Pi_u(t)$ among all nodes' subspaces at time t. So we are done. □

Assume now that Theorem 6.2 holds. To determine the tree structure, it is sufficient to determine the unique parent each node has. From the previous arguments, the parent of node u is the unique node v such that $\Pi_v(t)$ is the minimum dimension subspace that contains $\Pi_u(t)$. Then, the parent of node u is the node v such that

$$v = \underset{w \in V:\ d_{uw} = d_u}{\arg\min} d_w. \tag{6.8}$$

As we will discuss in Section 6.4.3, collecting the subspace information from the network nodes can be implemented efficiently. The algorithm that determines the tree topology reduces this information to only two "sufficient statistics": the dimension of each subspace $d_u = \dim(\Pi_u),\ \forall u \in V$, and the dimension of the intersection of every two subspaces $d_{uv} = \dim(\Pi_u \cap \Pi_v),\ \forall u, v \in V$, as described in Algorithm 6.2, assuming that the conditions of Theorem 6.2 hold.

Algorithm 6.2 Finding the network topology for a tree.

Input: $\{d_u\}$, $\{d_{uv}\}$
Output: the network topology $G(V, E)$

for all $u \in V$ **do**
  if $d_u = n$ **then**
    $u \leftarrow S$;
  else
    node u has as parent the node v with $v = \underset{w \in V:\ d_{uw} = d_u}{\arg\min} d_w$;
  end if
end for

6.4.1.2 Directed v.s. Undirected Network

In a tree with a single source, since new information can only flow from the source to each node along a single path, whether the network is directed or undirected makes no difference. In other words, from (6.7), all vectors that a node will send to its predecessor will belong in the subspace the predecessor already has. Thus Theorem 6.2 still holds for undirected networks with a common min-cut.

6.4.1.3 Different Min-Cuts

Assume now that the edges of the tree have different capacities, i.e., assigned different rates. In this case, the proof of Theorem 6.2 still holds, provided that

the condition in Theorem 6.2 is modified to

$$2D(G) < t < \lfloor \frac{n}{c_{\max}} \rfloor, \tag{6.9}$$

where $c_{\max} = \max_{v \in V} c_v$.

We underline that this theorem would not hold without the assumption in (6.1). Without this condition, it is possible that we cannot distinguish between nodes at same level with a common parent as explained in the following example.

Example 6.2. *If in the network in Figure 6.2, edge SA has unit capacity, while edge AB and AC have capacity two. In this case it is easy to see that there exists t_0 such that $\Pi_B(t) = \Pi_C(t) = \Pi_A(t-1)$, $\forall t \geq t_0$. Clearly in this case, we cannot distinguish between nodes B and C with this dissemination protocol.*

6.4.2 General Topologies

Consider now an arbitrary network topology, corresponding to a directed acyclic graph. An intuition we can get from examining tree structures is that, we can distinguish between two topologies provided all node subspaces are distinct. This is used to identify the unique parent of each node. In general topologies, it is similarly sufficient to identify the parents of each node, in order to learn the graph topology. The following theorem claims that having distinct subspaces is in fact a sufficient condition for topology identifiability over general graphs as well.

Theorem 6.3. *In a synchronous network employing randomized NC over $\mathbb{F}_q$, a sufficient condition to uniquely identify the topology with high probability as $q \gg 1$, is that*

$$\Pi_u(t) \neq \Pi_v(t) \quad \forall\, u, v \in V, \quad u \neq v, \tag{6.10}$$

for some time t. Under this condition, we can identify the topology by collecting global information at times t and $t+1$, i.e., two consecutive static views of the network.

Proof. Assume node u has the p parents $P(u) = \{u_1, \ldots, u_p\}$. Let

$$\Pi_u^{(u_1)}(t), \ldots, \Pi_u^{(u_p)}(t), \tag{6.11}$$

denote the subspaces node u has received from its parents up to time t, where $\Pi_u(t) = \sum_{i=1}^{p} \Pi_u^{(u_i)}(t)$. From construction it is clear that $\Pi_u^{(u_i)}(t+1) \sqsubseteq \Pi_{u_i}(t)$.

To identify the network topology, it is sufficient to decide which node $v \in V$ is the parent that sent the subspace $\Pi_u^{(u_i)}(t)$ to node u for each i, and thus find the p parents of node u. We claim that, provided (6.10) holds, node u has as parent the node v which at time t has the smallest dimension subspace containing $\Pi_u^{(u_i)}(t+1)$. Thus we can uniquely identify the network topology, by two static views, at times t and $t+1$, as Algorithm 6.3 describes.

Indeed, let $\pi_u^{(u_i)}(t)$ denote the subspace that node u receives from parent u_i at exactly time t, that is, $\Pi_u^{(u_i)}(t+1) = \Pi_u^{(u_i)}(t) + \pi_u^{(u_i)}(t+1)$. For each $i \in \{1, \ldots, p\}$, if $\pi_u^{(u_i)}(t+1) \nsubseteq \Pi_v(t)$ for all $v \in V \setminus \{u_i\}$, clearly $\Pi_u^{(u_i)}(t+1) \nsubseteq \Pi_v(t)$ for all $v \in V \setminus \{u_i\}$, and we are done. Otherwise, using Lemma 2.9 and because (6.10) holds, with high probability we have $\pi_u^{(u_i)}(t+1) \nsubseteq \Pi_v(t)$ for all $v \in V$ except those nodes that their subspaces contain $\Pi_{u_i}(t)$. So we are done. □

Note that to identify the network topology, we need to know, for all nodes u, the dimension $d_u \triangleq \dim(\Pi_u(t))$ of their observed subspaces at time t, the dimension $d_u^{(i)} \triangleq \dim(\Pi_u^{(u_i)}(t+1))$ for all parents u_i of node u, and the dimension of the intersection of $\Pi_u^{(u_i)}(t+1)$ with all $\Pi_w(t)$, $w \in V$, denoted as $d_{wu}^{(i)} \triangleq \dim(\Pi_u^{(u_i)}(t+1) \cap \Pi_w(t))$. Algorithm 6.3 uses this information to infer the topology.

Algorithm 6.3 Finding the topology of a general network.

Input: $\{d_u\}$, $\{d_u^{(i)}\}$, $\{d_{wu}^{(i)}\}$
Output: the network topology $G(V, E)$

for all $u \in V$ **do**
  if $d_u = n$ **then**
    $u \leftarrow S$;
  else
    for all $i \in \{1, \ldots, p_u\}$ **do**
      node u has as parent the node v with $v = \underset{w \in V:\ d_{wu}^{(i)} = d_u^{(i)}}{\arg\min} d_w$;
    end for
  end if
end for

The sufficient conditions (6.10) in Theorem 6.3, may or may not hold, depending on the network topology and the information dissemination protocol. Next, we will investigate for what network topologies the conditions (6.10) hold for the dissemination Algorithm 6.1 so that the network is identifiable.

Lemma 6.3. *Consider two arbitrary nodes u and v, where $P(u) = \{u_1, \ldots, u_{p_u}\}$ and $P(v) = \{v_1, \ldots, v_{p_v}\}$ are the parents of u and v respectively. Let $\Pi_{P(u)}(t-1) = \sum_{i=1}^{p_u} \Pi_{u_i}(t-1)$, and $\Pi_{P(v)}(t-1) = \sum_{i=1}^{p_v} \Pi_{v_i}(t-1)$. If $\Pi_u(t) = \Pi_v(t)$ we should have had $\Pi_{P(u)}(t-1) = \Pi_{P(v)}(t-1)$ w.h.p.*

Proof. Let us assume that $\Pi_u(t) = \Pi_v(t) = \Pi$. This implies that if $\pi_u(t)$ and $\pi_v(t)$ are subspaces collected by nodes u and v at time t then,

$$\Pi_u(t) = \Pi_v(t) = \Pi$$
$$\pi_u(t) + \Pi_u(t-1) = \pi_v(t) + \Pi_v(t-1).$$

From construction, we have $\Pi = \Pi_u(t) \sqsubseteq \Pi_{P(u)}(t-1)$ and $\Pi = \Pi_v(t) \sqsubseteq \Pi_{P(v)}(t-1)$.

On the other hand, since for every i, we randomly chose $\pi_u^{(u_i)}(t)$ from $\Pi_{u_i}(t-1)$ and since $\pi_u^{(u_i)}(t) \sqsubseteq \Pi$ (because $\pi_u(t) \sqsubseteq \Pi$) using Lemma 2.9 we conclude that we should have that $\Pi_{u_i}(t-1) \sqsubseteq \Pi$ which means we should have $\Pi_{P(u)}(t-1) \sqsubseteq \Pi$. Similarly, we should have $\Pi_{P(v)}(t-1) \sqsubseteq \Pi$. As a result, with high probability, we have to have

$$\Pi_{P(u)}(t-1) = \Pi_{P(v)}(t-1) = \Pi,$$

and we are done. □

Corollary 6.3. *If $\Pi_u(t) = \Pi_v(t) = \Pi$ for $t > l$ we should have had $\Pi_{P^l(u)}(t-l) = \Pi_{P^l(v)}(t-l) = \Pi$, w.h.p.*

Proof. Consider the parents of nodes u and v as super-nodes $P(u)$ and $P(v)$. Using a similar argument as stated in Lemma 6.3, we can conclude that the parents of $P(u)$ and $P(v)$, denoted as $P^2(u)$ and $P^2(v)$, should satisfy

$$\Pi_{P^2(u)}(t-2) = \Pi_{P^2(v)}(t-2) = \Pi.$$

We use this argument l times to get the result. □

Lemma 6.4. *If the dissemination protocol is in the steady state, $t \geq T_s$, we could not have $\Pi_u(t) = \Pi_v(t)$ unless nodes u and v have the same set of ancestors at some l level above in the network.*

Proof. Because $t \geq T_s$, we have $d_u(t) = \dim(\Pi_u(t)) < n$ and $d_v(t) = \dim(\Pi_v(t)) < n$. Let us assume $\Pi_u(t) = \Pi_v(t) = \Pi$ so we have $d \triangleq d_u(t) = d_v(t)$. From the Corollary 6.3 we can write

$$\Pi_{P^l(u)}(t-l) = \Pi_{P^l(v)}(t-l) = \Pi,$$

for every $l \geq 1$. Increasing l, two cases may happen. First, either $P^l(u)$ or $P^l(v)$ contains the source node S that results in $\dim(\Pi_{P^l(u)}(t-l)) = n$ or $\dim(\Pi_{P^l(v)}(t-l)) = n$ which is a contradiction since $d < n$. Second, nodes u and v have the same set of ancestors at some level l. □

Up to here, we have shown that assuming the dissemination protocol is in the steady state the subspaces of two arbitrary nodes are equal only if they have the same ancestors at some level above in the network. The following result, Theorem 6.4 states sufficient conditions that make the nodes' subspaces different for dissemination Algorithm 6.1.

Theorem 6.4. *Suppose two arbitrary nodes u and v have the same set of parents $P^l = P^l(u) = P^l(v)$ at some level l. The following conditions are sufficient so that the dissemination Algorithm 6.1 satisfies condition (6.10)* [4]*:*

$$\hat{c}_u \triangleq \text{min-cut}(P^l, u) \leq \text{min-cut}(S, P^l) \triangleq c_p,$$
$$\hat{c}_v \triangleq \text{min-cut}(P^l, v) \leq \text{min-cut}(S, P^l) \triangleq c_p.$$

4. Note that for the the min-cut c_u to node u, $c_u \triangleq \text{min-cut}(S, u)$, we have $c_u = \min[\hat{c}_u, c_p]$.

Proof. Consider the set of nodes in P^l. From the definition we know that there exists at least one path of length l from each node in P^l to the node u. But also there might exist paths of length less than l from some nodes in P^l to u. If this is the case, because the topology is a directed acyclic graph, we can find a subset P' of the nodes in P^l such that it forms a cut for the node u and the shortest path from each node in P' to u is l; see Figure 6.3. Moreover, we have min-cut$(S, P') = c_p$ and min-cut$(P', u) = \hat{c}_u$.

Now assume that $P' = \{p_1, \dots, p_k\}$ such that $\tau_{p_1} \leq \cdots \leq \tau_{p_k}$. Let $a_1, \dots, a_k$, be the accumulative min-cut from S to each node in P'. By this we mean that $a_1 = c_{p_1}$ and a_2 is the amount of increase in the min-cut from S by adding node p_2 and so on. We similarly consider the accumulative min-cut values from p_i to u and denote these by $b_1, \dots, b_k$. So we have $\sum_{j=1}^{k} a_j = c_p$ and $\sum_{j=1}^{k} b_j = \hat{c}_u$.

From definition of the waiting times (see Definition 6.1) we can write

$$d_{P'}(\tau_1) \geq a_1 + 1, \tag{6.12}$$

$$d_{P'}(\tau_2) \geq d_{P'}(\tau_1) + (\tau_2 - \tau_1)a_1 + a_2, \tag{6.13}$$

$$d_{P'}(\tau_k) \geq d_{P'}(\tau_{k-1}) + (\tau_k - \tau_{k-1}) \sum_{j=1}^{k-1} a_j + a_k. \tag{6.14}$$

Then we have

$$\begin{aligned} d_{P^l}(\tau_k) &\geq d_{P'}(\tau_k) \\ &\geq (\tau_2 - \tau_1)a_1 + \cdots + (\tau_k - \tau_{k-1}) \sum_{j=1}^{k-1} a_j + \sum_{j=1}^{k} a_j + 1. \end{aligned} \tag{6.15}$$

For d_u we can also write

$$d_u(\tau_1 + l) \leq b_1, \tag{6.16}$$

$$d_u(\tau_2 + l) \leq d_u(\tau_1 + l) + (\tau_2 - \tau_1) \min[a_1, b_1] + b_2, \tag{6.17}$$

$$d_u(\tau_k + l) \leq d_u(\tau_{k-1}) + (\tau_k - \tau_{k-1}) \min \left[\sum_{j=1}^{k-1} a_j, \sum_{j=1}^{k-1} b_j \right] + b_k, \tag{6.18}$$

or

$$\begin{aligned} d_u(\tau_k + l) \leq{}& (\tau_2 - \tau_2) \min[a_1, b_1] \\ &+ \cdots + (\tau_k - \tau_{k-1}) \min \left[\sum_{j=1}^{k-1} a_j, \sum_{j=1}^{k-1} b_j \right] + \sum_{j=1}^{k} b_j. \end{aligned} \tag{6.19}$$

From (6.15), (6.19) and the theorem assumptions we conclude that $d_u(\tau_k +$

$l) < d_{P^l}(\tau_k)$. Now for Δt time-slots later we write

$$\begin{aligned} d_u(\tau_k + l + \Delta t) &\overset{(a)}{\leq} d_u(\tau_k + l) + \hat{c}_u \Delta t \\ &\overset{(b)}{<} d_{P^l}(\tau_k) + c_p \Delta t \\ &\overset{(c)}{=} d_{P^l}(\tau_k + \Delta t), \end{aligned} \tag{6.20}$$

where (a) is true because u receives packets from P^l with rate at most $\hat{c}_u$; (b) is true because $d_u(\tau_k + l) < d_{P^l}(\tau_k)$ and $\hat{c}_u \leq c_p$; and finally (c) is true because after τ_k all of the nodes in P' receive packets at rate equal to their min-cut which means that P' (the same is true for P^l) receives packets at rate equal to its min-cut c_p.

The same inequality holds for the dimension of $\Pi_v(\tau_k + l + \Delta t)$. Thus for time $t > \tau_k + l$ we cannot have $\Pi_{P^l}(t - l) = \Pi_u(t)$ and $\Pi_{P^l}(t - l) = \Pi_v(t)$ if $\hat{c}_u \leq c_p$ and $\hat{c}_v \leq c_p$. So using Corollary 6.3 we are done. □

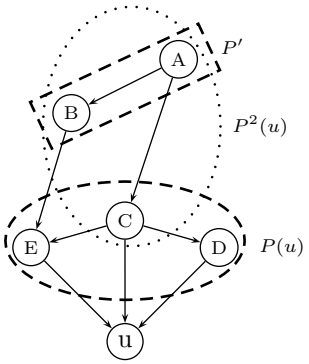


Figure 6.3 – Sets used in the proof of Theorem 6.4: the set $P(u)$ contains the parents of node u at distance $l = 1$; the set $P^2(u)$ contains the set of parents at distance $l = 2$; while P' is the subset of $P^2(u)$ at distance no less than $l = 2$.

Intuitively, what Theorem 6.4 tell us is that, if for a node u there exists a path that does not belong in any cut between the source and another node v, then nodes u and v will definitely have distinct subspaces. The only case where nodes u and v may have the same subspace is, if they have a common set of parents, a common cut. Even then, they would need both of them to receive all the innovative information that flows through the common cut at the same time. Note that the condition of Theorem 6.4 are also necessary for identifiably for the special case of tree topologies, such as the topology in Figure 6.2.

6.4.3 Practical Considerations

We here argue that our proposed scheme can lead to a practical protocol, where nodes passively collect information during the dissemination process,

and send once a small amount of information to the central node in charge of the topology inference. In particular, we assume that the nodes follow the information dissemination protocol and at some point the central node query them to report the subspaces they gather at a specific [5] time t.

We now calculate the communication cost (total number of bits required to be transmitted to a central node) of the proposed passive inference algorithm. Each node has to transmit at most $2\Delta_{\mathrm{i}}(G)$ subspaces to the central node where $\Delta_{\mathrm{i}}(G)$ is the maximum in-degree of nodes in the network. There are ϑ nodes in the network so $2\vartheta\Delta_{\mathrm{i}}(G)$ subspace have to be transmitted. The total number of subspaces of Π_S (which itself is an n-dimensional space) is

$$\sum_{i=1}^{n} \begin{bmatrix} n \\ i \end{bmatrix}_q \approx \sum_{i=1}^{n} q^{i(n-i)} \approx q^{n^2/4}, \tag{6.21}$$

where $\begin{bmatrix} n \\ i \end{bmatrix}_q$ is the Gaussian number defined in Section 2.3. To approximate the Gaussian number we use Lemma 2.1; note that the approximation holds for large q.

So to encode one of the subspace of Π_S we need approximately $\frac{n^2}{4}\log q$ bits. As a result, the total number of bits need to be transmitted to the central node is at most

$$\frac{2n^2\Delta_{\mathrm{i}}(G)\vartheta}{4}\log q. \tag{6.22}$$

Clearly, the complexity depends on the size of n, the number of packets that the source transmits. Here we assume that n is large enough, so that the network enters in steady state; on the other hand, other considerations such as decoding complexity at network nodes, would require n to take moderate values. Note that, for our algorithm to work, (i.e., to sample the network while in the steady state) we only require that $n = 2\beta c_{\max} D(G)$ (Corollary 6.1), where $\beta > 1$ is some constant that determines how many time slots the network is in the steady state. If n has such a size, the maximum number of bits that need to be transmitted per node (communication cost per node) is

$$R_{\text{com-cost/ND}} \approx 2\beta^2 c_{\max}^2 D(G)^2 \Delta_{\mathrm{i}}(G)\log q \quad \text{bits}. \tag{6.23}$$

In the above equation β, $c_{\max}$, and $\Delta_{\mathrm{i}}(G)$ are some constants. The only parameter that depends on the network size is $D(G)$. However for the most of practical content distribution networks the longest path of network is kept small to ensure a good connectivity between nodes in the network (see for example [81]).

To give a specific example for a possible communication cost, let us consider a practical scenario where $q = 2^8$, $c_{\max} = 1$, $\beta^2 = 5$, $\Delta_{\mathrm{i}}(G) = 5$, and

5. We assume the query is send before time t actually occurs; Also note that if the number of source packets n is much larger than the min-cut to each node, and if we have an estimate for $\Delta_{\mathrm{i}}(G)$ (the maximum in-degree of nodes in the network), a central node can with high probability select at time t in steady state. A node can also send a feedback message to inform the central node if it is not at steady state at time t.

$D(G) = 10$. Then we have $R_{\text{com-cost/nd}} \approx 4$ kilobytes. In contrast, in a practical dissemination scenario (e.g., video streaming) we would disseminate a large number of information packets each possibly as large as a few megabytes; thus the overhead of the topological information would not be significant.

6.5 Locating Byzantine Attackers

In this section we explore a problem that is dual to topology inference: given complete knowledge of the topology, we leverage subspace properties to identify the location of a malicious Byzantine attacker.

In a network coded system, the adversarial nodes in the network disrupt the normal operation of the information flow by inserting erroneous packets into the network. This can be done by inserting spurious data packets into their outgoing edges. One way in which these erroneous packets can be prevented from disrupting information flow is by reducing the transmission rate to below the min-cut of the network, and using the redundancy to protect against errors; [70, 71, 72]. One such technique, using subspaces to code information was proposed in [1]. In this approach, the source sends a basis of the subspace corresponding to the message. In the absence of errors, the linear operations of the intermediate nodes do not alter the sent subspace, and hence the receiver decodes the message by collecting the basis of the transmitted subspace. A malicious attacker inserts vectors that do not belong in the transmitted subspace. Therefore, if the message codebook uses subspaces that are "far enough" apart (according to an appropriately defined distance measure), then one can correct these errors [1]. Note that in this technique, we do not need any knowledge of the network topology for the error correction mechanism. All that is needed is that the intermediate nodes do not alter the transmitted subspace (which can be done if they do linear operations).

The approach of this section to locating adversaries uses the framework developed in the previous sections, where it was shown that under randomized NC, the subspaces gathered by the nodes of the network provide information about the topology. Therefore, the basic premise in this section is to use the structure of the erroneous subspace inserted by the adversary to reveal information about its location, when we already know the network topology.

6.5.1 Problem Formulation

Consider a network represented as a directed acyclic graph $G = (V, E)$. We have a source, sending information to r receivers, and one (or more) Byzantine adversaries, located at intermediate nodes of the network. We assume complete knowledge of the network topology, and consider the source and the receivers to be trustworthy (authenticated) nodes, that are guaranteed not to be adversaries.

Suppose source S sends n vectors, that span an n-dimensional subspace Π_S of the space $\mathbb{F}_q^L$, where we assume $q \gg 1$. In particular, in this section we will

consider (without loss of generality) subspace coding, where Π_S belongs to a codebook $\mathcal{C}$, $\Pi_S \in \mathcal{C}$ designed to correct network errors and erasures [1].

In the absence of any adversaries in the network each receiver R_i, $i = 1, \ldots, r$, can decode the exact space Π_S. Now assume that there is an adversary, Eve, who attacks one of the nodes in the network by combining a δ-dimensional subspace Π_E with its incoming space and sending the resulting vectors to its children. Then the receiver R_i collects some linearly independent vectors that span a subspace Π_{R_i}. We can write

$$\Pi_{R_i} = \mathcal{H}_i(\Pi_S + \Pi_E), \tag{6.24}$$

where $\mathcal{H}_i(\Pi)$ is a linear operator. This operator models the linear transformation that the network induces on the inserted source and adversary packets.

We assume that the receiver is able to at least detect that a Byzantine attack is under way. Moreover, we assume that the receiver is able to decode the subspace Π_S that the source has sent. This might be, either because the receiver has correctly decoded the sent message (i.e., using code construction from [1]), or, because after detecting the presence of an attack has requested the source subspace through a secure channel from the source node.

We can restrict the Byzantine attack in several ways, depending on the edges where the attack is launched, the number of corrupted vectors inserted, and the vertices (network nodes) that the adversary has access to. In this section we will distinguish between the cases where

I. there is a single Byzantine attacker located in a vertex of the network, and

II. there are multiple independent attackers, located on different vertices, that act without coordinating with each other.

Moreover, we assume that each attacker located on a single vertex is able to corrupt any outgoing edges by inserting arbitrary erroneous information.

Now, we are interested in understanding under what conditions we can uniquely identify the attacker's location (or, up to what uncertainty we can identify the attacker), under the above scenarios.

6.5.2 The Case of a Single Adversary

In this section we focus on the case where we want to locate a Byzantine adversary, Eve, controlling a *single* vertex of the network graph.

In Section 6.5.2.1 we illustrate the limitation of using *only* the information the receivers have observed along with the knowledge of the topology, to locate the adversary. This motivates requiring additional information from the intermediate nodes related to the subspaces observed by them. In Section 6.5.2.2, we show that such additional information allows us to localize the adversary either uniquely or within an ambiguity of at most two nodes.

6.5.2.1 Identification using only Topological Information

In order to illustrate the ideas, we will examine the case where the corrupted packets are inserted on a single edge of the network, say edge e_A. The extension to the cases where multiple edges get corrupted is easy.

Since each receiver R knows the subspaces $\{\Pi_R^{(i)}\}$ it has received from its $|\operatorname{In}(R)|$ parents, it knows whether what it received is corrupted or not (a subspace of Π_S or not). Using this, we can infer some information regarding topological properties that the edge e_A should satisfy. In particular we have the following result, Lemma 6.5.

Lemma 6.5. *Let P_e denote the set of paths*[6] *starting from the source and ending at edge e. Then, if $\mathcal{E}_C$ is the set of incoming edges to receivers that bring corrupted packets, while $\mathcal{E}_S$ the set of incoming edges to receivers that only bring source information, the edge e_A belongs in the set of edges $\mathcal{E}_A$, with*

$$\mathcal{E}_A \triangleq \left\{ \bigcap_{e\in\mathcal{E}_C} P_e - \bigcup_{e\in\mathcal{E}_S} P_e \right\}.$$

Proof. If R receives corrupted vectors from an incoming edge e then there exists at least one path that connects e_A to e. Then e_A is part of at least one path in P_e.

Conversely, if a receiver R does not receive corrupted packets from an incoming edge e, then e_A does not form part of any path in P_e. That is, there does not exist a path that connects e_A to e. □

The following example illustrates this approach.

Example 6.3. *Consider the network in Figure 6.4, and assume that R_1 receives corrupted packets from edge DR_1 and uncorrupted packets from AR_1, while R_2 receives only uncorrupted packets. Then $\mathcal{E}_A = \{DR_1\}$ and the attacker is located on node D.* ■

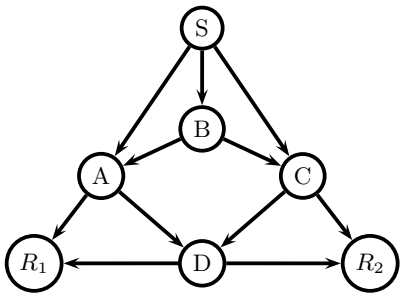


Figure 6.4 – The source S distributes packets to receivers R_1 and R_2.

6. In the following, we are going to equivalently think of P_e as the set of all edges that take part in these paths.

In Example 6.3, we were able to exactly identify the location of the adversary, because the set $\mathcal{E}_A$ contained a single edge, and node R_1 is trustworthy. It is easy to find network configurations where $\mathcal{E}_A$ contains multiple edges, or in fact all the network edges, and thus we can no longer identify the attacker. The following example illustrates one such case.

Example 6.4. *Consider the line network shown in Figure 6.5. Suppose the attacker is node A. If the receiver R sees a corrupted packet, then using just the topology, the attacker could be* any *of the other nodes in the line network. This illustrates that just the topology and receiver information could lead to large ambiguity in the location of the attacker.*

Therefore, Example 6.4 motivates the ideas examined in Section 6.5.2.2 which obtain additional information and utilize the structural properties of the subspaces observed.

6.5.2.2 Identification using Information from all Network Nodes

We will next discuss algorithms where a central authority, which we will call *controller*, requests from all nodes in the network to report some additional information, related to the subspaces they have received from their parents. The adversary could send inaccurate information to the controller, but the other nodes report the information accurately. Our task is to design the question to the nodes such that we can locate the adversary, despite its possible misdirection.

The controller may ask the nodes of the following types of information, listed in decreasing order of complexity:

Information 1: Each node v sends all subspaces $\Pi_v^{(i)}$ it has received from its parents, where $\Pi_v = \sum_{i \in P(v)} \Pi_v^{(i)}$.

Information 2: Each node v sends a randomly chosen vector from each of the received subspaces $\Pi_v^{(i)}$ ($|\operatorname{In}(v)|$ vectors in total).

Information 2 is motivated by the following observation made by Lemma 2.9: let Π_1 and Π_2 be two subspaces of $\mathbb{F}_q^n$, and assume that $\boldsymbol{y}$ be a randomly selected vector from Π_1. Then, for $q \gg 1$, $\boldsymbol{y} \in \Pi_2$ if and only if $\Pi_1 \sqsubseteq \Pi_2$. Thus, a randomly selected vector from Π_v allows to check whether $\Pi_v \sqsubseteq \Pi_S$ or not.

In fact, we will show in this section that for a single adversary it is sufficient to use [7] Information 2, and classify the edges of the network by simply testing whether the information flowing through each edge is a subspace of Π_S or not (i.e., is corrupted or not).

Theorem 6.5. *Using Information 1, by splitting the network edges into corrupted and uncorrupted sets, we can narrow the location of the adversary up to a set of at most two nodes. With Information 2, the same result holds w.h.p.*

7. Using Information 2, these statements are made with high probability, i.e., the probability goes to one as field size $q \to \infty$.

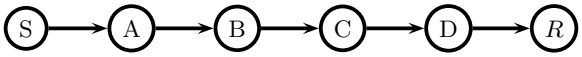


Figure 6.5 – The source S sends information to receiver R over a line network.

Proof. The network is a directed acyclic graph, so we can impose a partial order on the edges of the graph, such that $e_1 > e_2$ if e_1 is an ancestor edge of e_2 (i.e., there exists a path from e_1 to e_2). Then having Information 1 or Information 2, we can divide the edges of the network into two sets: the set of edges E_C through which are reported to flow corrupted subspaces, and the remaining edges E_S through which the source information flows so we have $E = E_S \cup E_C$ and $E_S \cap E_C = \emptyset$. Note that all the outgoing edges from the source belong in E_S.

Nodes in the network perform randomized NC so every node that receives corrupted information on at least one of its incoming edges makes all of the outgoing edges polluted w.h.p. Let t_v be the number of corrupted outgoing edges of a node v where we have $1 \leq t_v \leq |\operatorname{Out}(v)|$. For each node v that is not an adversary we have either $t_v = 0$ or $t_v = |\operatorname{Out}(v)|$.

Now, to prove the theorem we consider the following possible cases.

1. If the adversary Eve corrupts t_A outgoing edges where $1 < t_A < |\operatorname{Out}(A)|$ we can identify the node she has attacked uniquely because its behavior is different from all other nodes.
2. If she corrupts all of its outgoing edges, $t_A = |\operatorname{Out}(A)|$, then she can fraud us by declaring that one of the node's incoming edges is corrupted. If A declares more than one of the incoming edges as corrupted we can find its location uniquely.
3. She can also corrupt only one of its outgoing edges, $t_A = 1$, and pretends that its children is in fact the adversary by declaring all of its incoming edges bring non-corrupted information. She cannot declare that any of its incoming edges are polluted since then we may find its location uniquely.

In all of the above cases the adversary is on the boundary of two sets E_S and E_C and the ambiguity about its location is at most withing a set of two vertices where this set contains those two vertices that are connected by the corrupted edge with highest order among all corrupted edges (recall that we can compare all of the corrupted edges using the imposed partial order). □

6.5.3 The Case of Multiple Adversaries

In the case of a single adversary, it was sufficient to divide the set of edges into two sets, E_S and E_C, as described in the previous section. In the presence of multiple adversaries, this may no longer be sufficient. An additional dimension is that realistically, we may not know the exact number of adversaries present. In the following, we discuss a number of algorithms, that offer weaker or stronger identifiability guarantees.

6.5.3.1 Identification using only Topological Information

The approach in Section 6.5.2.1 can be directly extended in the case of multiple adversaries, but again, offers no identifiability guarantees.

Example 6.5. *Consider again the network in Figure 6.4, and assume that R_1 receives corrupted packets only from edge DR_1 while R_2 receives corrupted packets only from edge DR_2. Then $\mathcal{E}_A = \{AD, CD, DR_1, DR_2\}$ and (depending on our assumptions) we may have,*

- *a single adversary located on node D,*
- *two adversaries, located on nodes A and C,*
- *two adversaries, located on nodes A and D, or nodes C and D, or*
- *three adversaries, located on nodes A, C, and D.*

■

6.5.3.2 Identification using Splitting

Similar to Section 6.5.2.2, using Information 1 or Information 2, we can divide the set of edges into two sets E_S and E_C, depending on whether the information flowing through each edge belongs in Π_S or not. Depending on the network topology, we may be able to uniquely identify the location of the attackers. However, this approach, although it guarantees to find at least one of the attackers (within an uncertainty of at most two nodes), does not necessarily find all the attackers, even if we know their exact number.

To show this let us state the following definition.

Definition 6.3. *We say that node v is in the shadow of an adversary node A, if there exists a path that connects every incoming edge of v to a corrupted outgoing edge of A.*

Then we have the following result.

Lemma 6.6. *By splitting the network edges into two sets E_S and E_C we cannot identify adversarial nodes that are in the shadow of an adversary A.*

Proof. This is because if an attacker is in the shadow of another attacker, it may corrupt only already corrupted vectors and thus not incur a detectable effect. So we cannot distinguish between an attacker and a normal node that are in the shadow of A. □

The following example illustrates these points.

Example 6.6. *For the example in Figure 6.4, assume that each attacker corrupts all its outgoing edges, and consider the following two situations:*

1. *Assume that nodes A and C are attackers. If A reports truthfully while C lies we get $E_C = \{AD, AR_1, DR_1, DR_2, BC, CR_2, CD\}$, which allows to identify the attackers.*

2. *Assume that nodes B and D are attackers. Then we say that node D is in the shadow of node B, as it corrupts only already packets corrupted by B. Indeed, if $E_C = \{SB, BA, BC, AD, AR_1, DR_1, DR_2, BC, CR_2, CD\}$, knowing that the source is trustworthy, we can infer that node B is an attacker. However, any of the nodes A, C, and D can equally probably be the second attacker. All these nodes are in the shadow of node B.*

■

Theorem 6.6. *Using Information 1 it is possible to narrow down the location of those adversaries that have the highest order in the network using the splitting method. The same result holds for Information 2 w.h.p.*

Proof. As stated in the proof of Theorem 6.5 we can impose a partial order on the edges of the network graph. Then, by using Information 1 or Information 2 we may split the network edges into two sets E_S and E_C.

Because every node in the network performs randomized network coding, there are only two possibilities for each adversary to corrupt its outgoing edges and report subspaces for its incoming edges such that it is not located uniquely. These are as follows:

1. She corrupts some (or all) of its outgoing edges but reports its incoming edges as uncorrupted.
2. She corrupts all of its outgoing edges and reports some (at least one) of its incoming edges as corrupted.

Now, let us consider the set of all the corrupted edges that have highest order with respect to other corrupted edges and cannot be compared against each other. For each of the above cases there should be at least one adversary connected to every edge in this set. □

6.5.3.3 Identification using Subset Relationships

In this subsection we develop a new algorithm to find the adversaries which is based on Information 1.

For each node $u \in V$, let $P(u) = \{u_1, \ldots, u_{p_u}\}$ denote the set of parent nodes of u. We are going to treat $P(u)$ as a super-node, and use the notation $\Pi_{P(u)} = \sum_{i=1}^{p_u} \Pi_{u_i}$ for the union of the subspaces of all nodes in $P(u)$. Also recall that $\Pi_v^{(u)}$ denotes the subspace received by node v from node u.

Our last algorithm checks, for every node $u \in V$, whether

$$\Pi_v^{(u)} \stackrel{?}{\sqsubseteq} \Pi_{P(u)} \quad \forall v \in V : e_{uv} \in E. \tag{6.25}$$

Then we have the following result, Theorem 6.7.

Theorem 6.7. *If the pairwise distance between adversaries is greater than two, it is possible to find the exact number as well as the location of the attackers (within an uncertainty of parent-children sets) using the subset method.*

Proof. First, let us focus on a single adversary case where $A \in V$ is the node attacked by the adversary. Then we will generalize the idea for an arbitrary number of adversaries.

If (6.25) is satisfied for all children of u, we know that node u is not an adversary. If the relationship is not satisfied, that is $\Pi_v^{(u)} \nsubseteq \Pi_{P(u)}$ for at least one child of u, we consider node u as a potential candidate for being an adversary. For sure we know that

$$\Pi_v^{(A)} \nsubseteq \Pi_{P(A)} \quad \forall v \in V : e_{Av} \in E,$$

but depending on the subspace that the adversary reports, the relation (6.25) may not be also satisfied for other nodes. Based on what the adversary reports there would be two possible cases.

If the adversary pretends that it is a trustworthy node (just declares the received subspace from its parents) the above relation also fails for the children of A who receive corrupted subspaces. On the other hand, if the adversary tells the truth and declares its corrupted subspace, we have

$$\Pi_A^{(u)} \nsubseteq \Pi_{P(u)} \quad \forall u \in V : uA \in E.$$

Thus the ambiguity set we have identified includes the adversary and its parents and/or its children depending on the adversary's report.

Repeating this procedure for every node in the network, we can identify sets of potential adversaries. We know that depending on the adversaries action there exists ambiguity in finding their exact location. In fact in the worst case, the uncertainty is within a set of nodes including the adversary, its parents and its children. So if the distance between adversaries is greater than two, the "uncertainty" sets do not overlap. In this case we can easily distinguish between different adversaries. □

This procedure allows to identify adversaries (within the mentioned parent-children ambiguity set), even if one is in the shadow of another, and even if we do not know their exact number, provided they are "far enough" in the network to be distinguishable.

6.6 Practical Implications for Topology Management

In Section 6.4, we demonstrated that using subspaces of all nodes, we can infer the network topology under certain conditions. In this section, we will show that even from what a single node observes, it is possible to get some information regarding the bottlenecks and clustering in the network.

Leveraging this observation in the context of P2P networks, we propose algorithms that use this information in a distributed peer-initiated manner to avoid bottlenecks and clustering.

6.6.1 Problem Statement and Motivation

In peer-to-peer networks that employ NC for content distribution (see for example Avalanche [61, 62]) we want to create and maintain a well-connected network topology, to allow the information to flow fast between the nodes; however, this is not straightforward. Peer-to-peer networks are very dynamically changing networks, where hundreds of nodes may join and leave the network within seconds. All nodes in this network are connected to a small number of neighbors (e.g., four to eight). An arriving node is allocated neighbors among the active participating nodes [8], which accept the solicited connection unless they have already reached their maximum number of neighbors. As a result, nodes that arrive at around the same time tend to get connected to each other, since they are all simultaneously available and looking for neighbors. That is, we have formation of clusters and bottlenecks in the network.

To avoid this problem, one method adopted in protocols is to ask all nodes to periodically drop one neighbor and reconnect to a new one among an active peers list. This randomized rewiring results in a fixed average number of reconnections per node independently of how good or bad is the formed network topology. Thus to achieve a good, on the average, performance in terms of breaking clusters, it entails a much larger number of rewiring than required, and unnecessary topology changes.

An alternative approach is to have peers initiate topology rewirings when they detect they are in a cluster. Clearly a central node could keep some structural information, i.e., keep track of the current network topology, and use it to make more educated choices of neighbor allocations. However, the information this central node can collect only reflects the *overlay* network topology, and is oblivious to bandwidth constraints from the underlying physical links. Acquiring bandwidth information for the underlying physical links at the central node requires costly estimation techniques over large and heterogeneous networks, and steers towards a centralized network operation. We will argue that such bottlenecks can be inferred almost passively in a peer-initiated manner, thus alleviating these drawbacks.

Here, we will show that the coding vectors the peers receive from their neighbors can be used to passively infer bottleneck information. This allows individual nodes to initiate topology changes to correct problematic connections. In particular, peers by keeping track of the coding vectors they receive can detect problems in both the overlay topology and the underlying physical links. The following example illustrates these points.

Example 6.7. *Consider the toy network depicted in Figure 6.6(a) where the edges correspond to logical (overlay network) links. The source S has n packets to distribute to four peers. Nodes A, B and C are directly connected to the source S, and also among themselves with logical links, while node D is*

8. This is usually done by a central node which we call it (following Avalanche [61, 62]) "registrat". This is the central authority that keeps the list of all nodes in the network and gives every new node a set of neighbors.

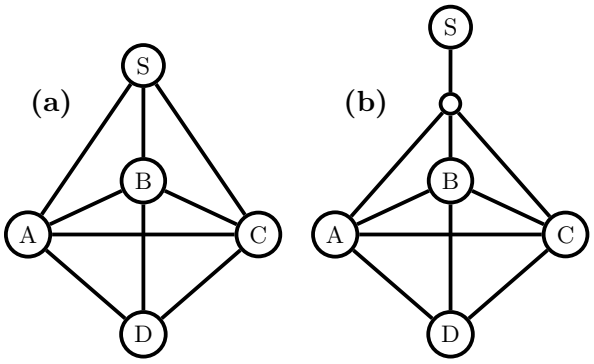


Figure 6.6 – The source S distributes packets to the peers A, B, C and D over the overlay network (a), that uses the underlying physical network (b).

connected to nodes A, B and C. In this overlay network, there exist three edge-disjoint paths between source and any other nodes.

Assume now (as shown in Figure 6.6(b)) that the logical links SA, SB, SC share the bandwidth of the same underlying physical link, which forms a bottleneck between the source and the remaining nodes of the network. As a result, assume the bandwidth on each of these links is only $1/3$ of the bandwidth of the remaining links. A central node (registrat), even if it keeps track of the complete logical network structure by querying each node asking about its neighbors, is oblivious to the existence of the bottleneck and the asymmetry between the link bandwidths.

Node D however, can infer this information by observing the coding vectors it receives from its neighbors A, B and C. Indeed, when node A receives a coded packet from the source, it will forward a linear combination of the packets it has already collected to nodes B and C and D. Now each of the nodes B and C, once they receive the packet from node A, they also attempt to send a coded packet to node D. But these packets will not bring new information to node D, because they will belong in the linear span of coding vectors that node D has already received. Similarly, when nodes B and C receive a new packet from the source, node D will end up being offered three coded packets, one from each of its neighbors, and only one of the three will bring to node D new information.
■

More formally, the coding vectors nodes A, B and C will collect will effectively span the same subspace; thus the coded packets they will offer to node D to download will belong in significantly overlapping subspaces and will thus be redundant (we formalize these intuitive arguments in Section 6.6.2). Node D can infer from this passively collected information that there is a bottleneck between nodes A, B, C and the source, and can thus initiate a connection change.

6.6.2 Theoretical Framework

Here we use the same notations introduced in Section 6.2. For simplicity we will assume that the network is synchronous [9]. Nodes are allowed to transmit linear combinations of their received packets only at clock ticks, at a rate equal to the adjacent link bandwidth.

Now we use the framework of Section 2.4 to investigate the information that we can obtain from the local information of a node's subspace. From notations defined in Section 6.2, we know that for an arbitrary node v we can write

$$\Pi_v(t) = \sum_{i \in P(v)} \Pi_v^{(i)}(t). \tag{6.26}$$

We are interested in understanding what information we can infer from these received subspaces $\Pi_v^{(i)}$, $i \in P(v)$, about bottlenecks in the network. For example, the overlap of subspaces from the neighbors reveals some information about bottlenecks. Therefore, we need to show that such overlaps occur due to topological properties and not due to particular random linear combinations chosen by the network code.

Let us assume that the subspaces $\Pi_v^{(i)}$ a node v receives from its set of parents $P(v)$ have an intersection of dimension d. Then we have the following observations.

Observation 6.1. *The subspaces $\Pi_v^{(i)}$, $i \in P(v)$, of the neighbors have an intersection of size at least d (see Corollary 2.1).*

Observation 6.2. *The min-cut between the set of nodes $P(v)$ and the source is smaller than the min-cut between the node v and set $P(v)$ (see Theorem 6.1).*

In the following, we will discuss algorithms that use such observations for topology management.

6.6.3 Algorithms

Our peer-initiated algorithms for topology management consist of three tasks:

1. Each peer decides whether it is satisfied with its connection or not, using a *decision criterion*.
2. An unsatisfied peer sends a *rewiring request*, that can contain different levels of information, either directly to the registrat, or to its neighbors (these are the only nodes the peer can communicate with).
3. Finally, the registrat, having received rewiring requests, *allocates neighbors* to nodes to be reconnected.

9. This is not essential for the algorithms but simplifies the theoretical analysis.

The decision criterion can capitalize on the fact that overlapping received subspaces indicate an opportunity for improvement. For example, in the first algorithm we propose (Algorithm 1), a node can decide it is not satisfied with a particular neighbor, if it receives $k > 0$, non-innovative coding vectors from it, where k is a parameter to be decided. Then it has each unsatisfied node directly contact the registrat and specify the neighbor it would like to change. The registrat randomly selects a new neighbor. This algorithm, as we demonstrate through simulation results, may lead to more rewirings than necessary: indeed, all nodes inside a cluster may attempt to change their neighbors, while it would have been sufficient for a fraction of them to do so.

Our second algorithm (Algorithm 2) uses a different decision criterion: for every two neighbors u and v, each peer w computes the rate at which the received joint space $\Pi_w^{(u)} + \Pi_w^{(v)}$ and intersection space $\Pi_w^{(u)} \cap \Pi_w^{(v)}$ increases. If the ratio between these two rates becomes greater than a threshold $\mathcal{T}$, the node decides it would like to change one of the two neighbors. However, instead of directly contacting the registrat, it uses a decentralized voting method that attempts to further reduce the number of reconnections. Then the registrat randomly selects and allocates one new neighbor for the nodes have sent rewiring request.

Our last proposed algorithm (Algorithm 3), while still peer-initiated and decentralized, relies more than the two previous ones in the computational capabilities of the registrat. The basic observation is that, nodes in the same cluster will not only receive overlapping subspaces from their parents, but moreover, they will end up collecting subspaces with very small distance (this follows from Theorem 6.1 and Corollary 2.1 and is also illustrated through simulation results in Section 6.6.4; see Figure 6.8). Each unsatisfied peer v sends a rewiring request to the registrat, indicating to the registrat the subspace Π_v it has collected. A peer can decide it is not satisfied using for example the same criterion as in Algorithm 2.

The registrat waits for a short time period, to collect requests from a number of dissatisfied nodes. These are the nodes of the network that have detected they are inside clusters. It then calculates the distance between the identified subspaces to decide which peers belong in the same cluster. While exact such calculations can be computationally demanding, in practice, the registrat can use one of the many hashing algorithms to efficiently do so. Finally the registrat breaks the clusters by rewiring a small number of nodes in each cluster. The allocated new neighbors are either nodes that belong in different clusters, or, nodes that have not send a rewiring request at all.

We will compare our proposed algorithms against the *Random Rewiring* currently employed by many peer-to-peer protocols (e.g., see [61, 62, 81]). In this algorithm, each time a peer receives a packet, with probability p contacts the registrat and asks to change a neighbor. The registrat randomly selects which neighbor to change, and randomly allocates a new neighbor from the active peer nodes.

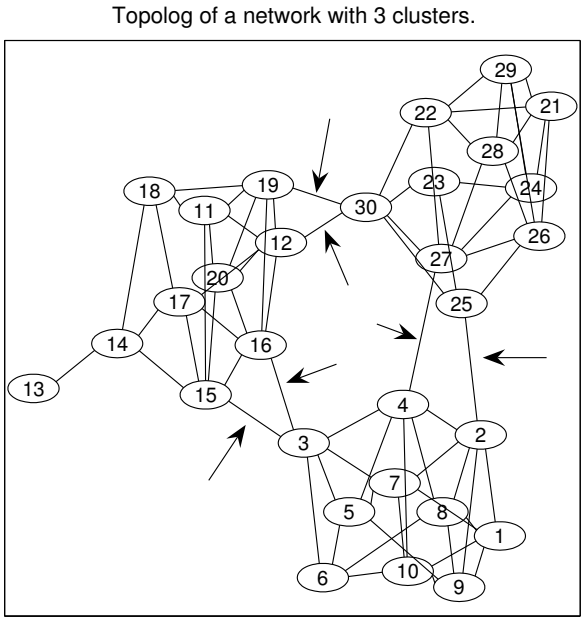


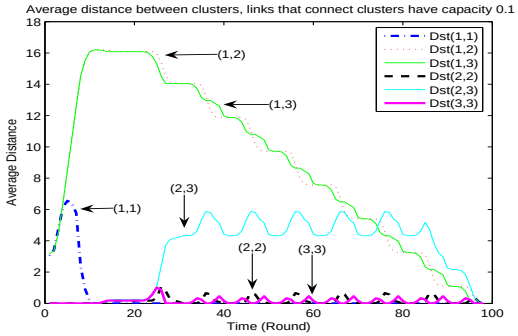
Figure 6.7 – A sample of topology with three clusters: cluster 1 contains nodes 1–10, cluster 2 nodes 11–20 and cluster 3 nodes 21–30.

6.6.4 Simulation Results

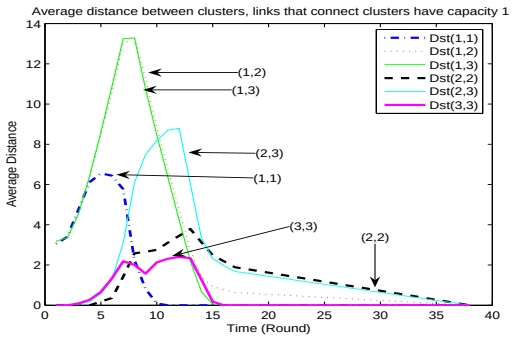
For our simulation results we will start from randomly generated topologies similar to Figure 6.7, that consists of 30 nodes connected into three distinct clusters. The source is node 1, and belongs in the first cluster. The bottleneck links are indicated with arrows (and thus indicate the underlying physical link structure). Our first set of simulation results depicted in Figure 6.8 show that the subspaces within each cluster are very similar, while the subspaces across clusters are significantly different, where we use the distance measure $D_S(\cdot,\cdot)$ defined in (2.5). These results indicate that knowledge of these subspaces will allow the registrat to accurately detect and break clusters (Algorithm 3).

Our second set of simulation results considers again topologies with three clusters: cluster 1 has 15 nodes and contains the source, cluster 2 has also 15 nodes, while the number of nodes in cluster 3 increases from 15 to 250. During the simulations we assume that the registrat keeps the nodes' degree between 2 and 5, with an average degree of 3.5. All edges correspond to unit capacity links.

We compare the performance of the three proposed algorithms in Section 6.6.3 with random rewiring. We implemented these algorithms as follows. For random rewiring, every time a node receives a packet it changes one of its neighbors with probability $p = \frac{8}{500}$. For Algorithm 1, we use a parameter



(a) Bottleneck links capacity is equal to 0.1.



(b) Bottleneck links capacity is equal to 1.

Figure 6.8 – Simulation results for the topology in Figure 6.7, with bottleneck links capacity value equal to 0.1 (top) and 1 (bottom).

of $k = 10$, and check whether the non-innovative packets received exceed this value every four received packets. For Algorithm 2, every node checks the ratio of the dimensions of the intersection and the joint space of subspaces received from each pair of neighbors using the threshold value $\mathcal{T} = 1$. Finally for Algorithm 3, we assume that nodes use the same criterion as in Algorithm 2 to decide whether they form part of a cluster, again with $\mathcal{T} = 1$. Dissatisfied nodes send their observed subspaces to the registrat. The registrat assigns nodes u and v in the same cluster if $d_S(\Pi_u, \Pi_v) \leq 7$.

Table 6.1 compares all algorithms with respect to the average collection time, defined as the difference between the time a peer receives the first packet and the time it can decode all packets, and averaged over all peers. All algorithms perform similarly, indicating that all algorithms result in breaking the clusters. It is important to note that the average collection time is in terms of number of exchanges needed and *does not* account for the delays incurred due to rewiring. We compare the number of such rewirings needed next.

Figure 6.9 plots the average number of rewirings each algorithm employs.

Random rewiring incurs a number of rewirings proportional to the number of P2P nodes, and independently from the underlying network topology. Our proposed algorithms on the other hand, adapt to the existence and size of clusters. Algorithm 3 leads to the smallest number of rewirings. Algorithm 2 leads to a larger number of rewirings, partly due to that the new neighbors are chosen randomly and not in a manner that necessarily breaks the clusters. The behavior of algorithm 1 is interesting. This algorithm rewires any node that has received more than k non-innovative packets. Consider cluster 3, whose size we increase for the simulations. If k is small with respect to the cluster size, then a large number of nodes will collect close to k non-innovative packets; thus a large number of nodes will ask for rewirings. Moreover, even after rewirings that break the cluster occur, some nodes will still collect linearly dependent information and ask for additional rewirings. As cluster 3 increases in size, the information disseminates more slowly within the cluster. Nodes in the border, close to the bottleneck links, will now be the ones to first ask for rewirings, long before other nodes in the network collect a large number of non-innovative packets. Thus once the clusters are broken, no new rewirings will be requested. This desirable behavior of Algorithm 1 manifests itself for large clusters; for small clusters, such as cluster 2, the second algorithm for example achieves a better performance using less reconnections.

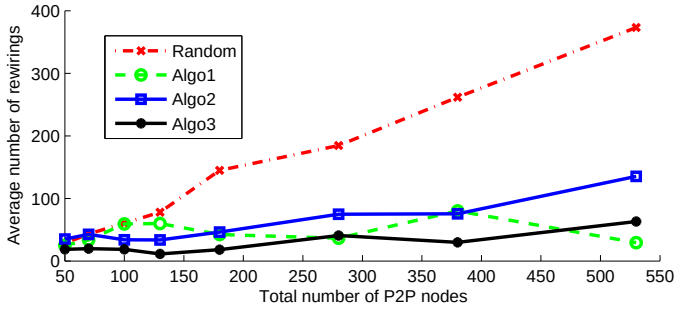


Figure 6.9 – Average number of rewirings, for a topology with three clusters: cluster 1 has 15 nodes, cluster 2 has 15 nodes, while the number of nodes in cluster 3 increases from 20 to 250 as described in Table 6.1.

Table 6.1 – Average Collection Time

Topology	Random	Algo 1	Algo 2	Algo 3
15–15–20	20.98	22.14	20.57	20.39
15–15–40	18.72	21.13	19.36	19.47
15–15–70	18.88	21.54	18.97	19.54
15–15–100	18.6	21.48	18.91	21.42
15–15–150	19.56	20.85	19.96	20.18
15–15–250	18.79	19.8	19.18	18.99

6.7 Concluding Remarks

In this chapter we explored the properties of subspaces each node collects in networks that employ randomized NC and found that there exists an intricate relationship between the structure of the network and these properties. This observation led us to utilize these relationships in several different applications. As the first application, we studied the conditions under which we can passively infer the network topology during content distribution. We showed that these conditions are not very restrictive and hold for a general class of information dissemination protocols. As our second application, which in some sense is the dual of the previous problem, we focused on locating Byzantine attackers in the network. We studied and formulated this problem and found that for the single adversary we can identify the adversary within an uncertainty of two nodes. For the case of multiple adversaries, we discussed a number of algorithms and conditions under which we can guarantee identifiability. For our last application, we investigated the relation between the bottlenecks in a logical network and the subspaces received at a specific network node. We leveraged our observations to propose decentralized peer-initiated algorithms for rewiring in P2P systems to avoid clustering in a cost-efficient manner, and evaluated our algorithms through simulations results.

The applications studied in this chapter demonstrate advantages of using randomized NC for network management and control, that are additional to throughput benefits. These are just a few examples and we believe that there exist a lot more applications where we can use the subspace properties developed in this chapter. We hope that these properties will become part of a toolbox that can be used to develop applications for systems that employ NC techniques.

6.A Omitted Proofs

Proof of Lemma 6.1. Here we assume that n is very large. Then in Corollary 6.1 we will derive a sufficient condition on the largeness of n.

Let v be the node that has the longest path to the source S. Because of Definition 6.1 we can write $T_s \leq \tau_v$. Then we may upper bound τ_v as follows

$$\tau_v \leq 2 + \max_{u \in P(v)} \tau_u, \tag{6.27}$$

where $P(v)$ is the set of parents of v. Now we can repeat the above argument until we reach the source S. So finally we have

$$\tau_v \leq 2D(G), \tag{6.28}$$

which leads to the lemma's assertion. □

Proof of Lemma 6.2. Let us write

$$\begin{aligned}
&\dim\big(\pi_u(1) \cap \Pi_v(j)\big) \\
&\quad \overset{(a)}{=} \dim\big(\pi_u(1) \cap (\Pi_v(j) \cap \Pi(0))\big) \\
&\quad \overset{(b)}{=} \dim\big(\pi_u(1) \cap \pi_v(1)\big) \\
&\quad \overset{(c)}{=} \min\big[d_0, (k_u(1) + k_v(1) - d_0)^+, k_u(1), k_v(1)\big] \\
&\quad = \big(k_u(1) + k_v(1) - d_0\big)^+ \\
&\quad < k_u(1),
\end{aligned} \tag{6.29}$$

where (a) follows because $\pi_u(1) \sqsubseteq \Pi(0)$ and (c) is a result of Corollary 2. So $\forall j \in \{1, \ldots, t\}$ we have $\pi_u(1) \nsubseteq \Pi_v(j)$ which results in $\Pi_u(i) \nsubseteq \Pi_v(j)$, $\forall i, j \in \{1, \ldots, t\}$. By symmetry, we have the second assertion of the lemma, namely, $\Pi_v(j) \nsubseteq \Pi_u(i)$, $\forall i, j \in \{1, \ldots, t\}$.

Now, it only remains to check (b). We will prove this by induction. Obviously, $\Pi(0) \cap \Pi_v(1) = \pi_v(1)$. Suppose that we have $\Pi(0) \cap \Pi_v(k) = \pi_v(1)$ where $k < t$ then we show that it also holds for $k + 1$.

We know that $\pi_v(1) \sqsubseteq \Pi(0) \cap \Pi_v(k+1)$. To show that $\Pi(0) \cap \Pi_v(k+1) \sqsubseteq \pi_v(1)$ we proceed as follows. Let $\boldsymbol{w} \in \Pi(0) \cap \Pi_v(k+1)$ then $\boldsymbol{w} \in \Pi(0)$ and $\boldsymbol{w} \in \Pi_v(k+1) = \sum_{i=1}^{k+1} \pi_v(i)$. We may decompose $\boldsymbol{w}$ as $\boldsymbol{w} = \sum_{i=1}^{k+1} \boldsymbol{w}_i$ where $\boldsymbol{w}_i \in \pi_v(i)$. Then note that $\boldsymbol{w}_{k+1} = \boldsymbol{w} - \sum_{i=1}^{k} \boldsymbol{w}_i \in \Pi(k-1)$ and by using Lemma 2.10 it can be shown that $\Pi(k-1) \cap \pi_v(k+1) = \emptyset$ w.h.p. So we conclude that $\boldsymbol{w}_{k+1} = 0$ which means $\boldsymbol{w} \in \Pi_v(k)$. This shows that $\boldsymbol{w} \in \Pi(0) \cap \Pi_v(k)$ where by induction assumption we have $\boldsymbol{w} \in \pi_v(1)$ and we are done. □

Proof of Corollary 6.2. Because $\Pi_u(0) \nsubseteq \Pi_v(j-1)$ then by Lemma 2.9 we have $\pi_a(1) \nsubseteq \Pi_v(j-1)$ w.h.p. So as a result we have $\Pi_a(i) \nsubseteq \Pi_v(j-1)$ $\forall i, j \in \{1, \ldots, t\}$. Now, because $\Pi_b(j) \sqsubseteq \Pi_v(j-1)$ we conclude that $\Pi_a(i) \nsubseteq \Pi_b(j)$ $\forall i, j \in \{1, \ldots, t\}$ w.h.p. By symmetry, we also deduce the other part of the corollary. □

6.B Algebraic Model for Synchronous Networks

In this appendix we employ an algebraic approach to analyze the dissemination protocol given in Algorithm 6.1. This approach is similar to [7] and [4], but differs in that we introduce memory into the coding process.

We introduce memory as follows. Suppose we are interested in finding the transfer function between the source and an arbitrary node v. Let $\boldsymbol{X}$ be a $n \times L$ matrix with rows the n packets (vectors) that the source wants to transmit to the receivers. We assume that $\dim(\langle \boldsymbol{X} \rangle) = n$. Let $\boldsymbol{Y}[t] \in \mathbb{F}_q^{\xi \times L}$ be a matrix with rows the packets that pass through the ξ edges of the network at time t. Let $\boldsymbol{Z}^{(v)}[t]$ be the set of packets that node v receives at time t. Similar to [7], we will write state-space equations that involve these vectors; however, we will ensure that, at each time t, coding at each node occurs across all the packets that the node has received before time t.

In every time-slot t, the source injects $|\operatorname{Out}(S)|$ packets into the network that are random linear combinations of the original source packets $\boldsymbol{X}$. These linear combinations can be captured as $\boldsymbol{M}[t]\boldsymbol{X}$, where $\boldsymbol{M}[t] \in \mathbb{F}_q^{|\operatorname{Out}(S)| \times n}$ is a random matrix. Intermediate network nodes will transmit packets on their outgoing edges depending on the network connectivity, and the state of the dissemination protocol.

The network connectivity can be captured by the $\xi \times \xi$ adjacency matrix $\mathcal{F}$ of the labeled line graph of the graph G, defined as follows

$$\mathcal{F}_{ij} \triangleq \begin{cases} 1 & \operatorname{head}(e_i) = \operatorname{tail}(e_j), \\ 0 & \text{otherwise.} \end{cases} \tag{6.30}$$

To model random coding over a finite field $\mathbb{F}_q$, we consider a sequence of random matrices $\boldsymbol{F}_1^{(t)}, \ldots, \boldsymbol{F}_{t-1}^{(t)}$ which conform to $\mathcal{F}$. That is, the entries of these matrices have for $i \neq j$ $(\boldsymbol{F}_k^{(t)})_{ij} = 0$ wherever $\mathcal{F}_{ij} = 0$ and have random numbers from $\mathbb{F}_q$ in all other places. The set of matrices $\boldsymbol{F}_1^{(t)}, \ldots, \boldsymbol{F}_{t-1}^{(t)}$ represent the network code at every time-slot t.

The dissemination protocol dictates when a node can start transmitting packets, according to its waiting time (equivalently, when the outgoing edges of the node will have packets send through them). To capture this, we will use the step function $u[t]$,

$$u[t] \triangleq \begin{cases} 1 & t \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{6.31}$$

and define the $\xi \times \xi$ diagonal matrix $\boldsymbol{U}[t]$ as,

$$\forall i \in E: \quad \boldsymbol{U}_{ii}[t] \triangleq u\left[t - \tau_{\operatorname{tail}(i)} - 1\right], \tag{6.32}$$

where τ_v is the *waiting time* for node v. In this section we assume that the waiting times may have arbitrary values and we do not restrict them according to Definition 6.1.

Using the above definitions, the set of packets (vectors) that each node v receives in every time instant $t > 0$ can be written as follows

$$\begin{cases} \boldsymbol{Y}[t] = \boldsymbol{U}[t] \left(\boldsymbol{A}\boldsymbol{M}[t]\boldsymbol{X} + \sum_{i=1}^{t-1} \boldsymbol{F}_i^{(t)} \boldsymbol{Y}[t-i] \right), \\ \boldsymbol{Z}^{(v)}[t] = \boldsymbol{B}^{(v)} \boldsymbol{Y}[t], \end{cases} \tag{6.33}$$

where $\boldsymbol{Y}[0] = \boldsymbol{0}$. In the above, $\boldsymbol{A} \in \mathbb{F}_q^{\xi \times |\operatorname{Out}(S)|}$ is a matrix which represents the connection of node S to the rest of the network. In the same way matrix $\boldsymbol{B}^{(v)} \in \mathbb{F}_q^{|\operatorname{In}(v)| \times \xi}$ defines the connection of node v to the set of edges in the network.

It is worth noting that although (6.33) is written for the packets transmitted on each edge, we can write the same set of equations for the coding vectors.

Suppose we are interested in finding the output of such a system at some time instant T. We can rewrite the above equations by defining new matrices as follows. We can collect the source random operations as

$$\boldsymbol{M}_T \triangleq \begin{bmatrix} \boldsymbol{M}[1] \\ \vdots \\ \boldsymbol{M}[T] \end{bmatrix} \in \mathbb{F}_q^{T|\operatorname{Out}(S)| \times n}. \tag{6.34}$$

For the states of system we define

$$\boldsymbol{Y}_T \triangleq \begin{bmatrix} \boldsymbol{Y}[1] \\ \vdots \\ \boldsymbol{Y}[T] \end{bmatrix} \in \mathbb{F}_q^{\xi T \times L}. \tag{6.35}$$

We also define a new set of matrices which represent the input-output relation. Using matrix $\boldsymbol{A}$ we define the following matrix

$$\boldsymbol{A}_T \triangleq \boldsymbol{I}_T \otimes \boldsymbol{A} = \begin{bmatrix} \boldsymbol{A} & & \\ & \ddots & \\ & & \boldsymbol{A} \end{bmatrix} \in \mathbb{F}_q^{\xi T \times T|\operatorname{Out}(S)|}. \tag{6.36}$$

For the connection of node v we define

$$\boldsymbol{B}_T^{(v)} \triangleq \begin{bmatrix} \boldsymbol{0}_{|\operatorname{In}(v)| \times (T-1)\xi} & \boldsymbol{B}^{(v)} \end{bmatrix} \in \mathbb{F}_q^{|\operatorname{In}(v)| \times \xi T}. \tag{6.37}$$

We define matrix $\boldsymbol{F}_T$ which represent how the states are related to each other

$$\boldsymbol{F}_T \triangleq \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots \\ \boldsymbol{F}_1^{(2)} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots \\ \boldsymbol{F}_2^{(3)} & \boldsymbol{F}_1^{(3)} & \boldsymbol{0} & \boldsymbol{0} & \cdots \\ \boldsymbol{F}_3^{(4)} & \boldsymbol{F}_2^{(4)} & \boldsymbol{F}_1^{(4)} & \boldsymbol{0} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \in \mathbb{F}_q^{\xi T \times \xi T}. \tag{6.38}$$

Finally, we use matrix $\boldsymbol{U}_T$ that captures the time when transmissions start for each edge

$$\boldsymbol{U}_T \triangleq \begin{bmatrix} \boldsymbol{U}[1] & & \\ & \ddots & \\ & & \boldsymbol{U}[T] \end{bmatrix} \in \mathbb{F}_q^{\xi T \times \xi T}. \tag{6.39}$$

Using the above definitions, we can rewrite (6.33) as follows

$$\begin{cases} \boldsymbol{Y}_T = \boldsymbol{U}_T \left(\boldsymbol{A}_T \boldsymbol{M}_T \boldsymbol{X} + \boldsymbol{F}_T \boldsymbol{Y}_T \right), \\ \boldsymbol{Z}^{(v)}[T] = \boldsymbol{B}_T^{(v)} \boldsymbol{Y}_T. \end{cases} \tag{6.40}$$

This equation can be solved to find the input-output transfer matrix at time T which results in

$$\boldsymbol{Z}^{(v)}[T] = \underbrace{\left[\boldsymbol{B}_T^{(v)} (\boldsymbol{I} - \boldsymbol{U}_T \boldsymbol{F}_T)^{-1} \boldsymbol{U}_T \boldsymbol{A}_T \boldsymbol{M}_T \right]}_{\boldsymbol{H}^{(v)}[T]} \boldsymbol{X}, \tag{6.41}$$

where $\boldsymbol{H}^{(v)}[T] \in \mathbb{F}_q^{|\operatorname{In}(v)| \times n}$. From the definition of matrix $\boldsymbol{F}_T$, we know that it is a "strictly lower triangular matrix" which means $\boldsymbol{F}_T$ is nilpotent and we have $\boldsymbol{F}_T^T = 0$. The same applies for the matrix $\boldsymbol{U}_T \boldsymbol{F}_T$, namely we have $(\boldsymbol{U}_T \boldsymbol{F}_T)^T = 0$. So the matrix $(\boldsymbol{I} - \boldsymbol{U}_T \boldsymbol{F}_T)^{-1}$ has an inverse which is equal to

$$(\boldsymbol{I} - \boldsymbol{U}_T \boldsymbol{F}_T)^{-1} = \boldsymbol{I} + \cdots + (\boldsymbol{U}_T \boldsymbol{F}_T)^{T-1}. \tag{6.42}$$

Finally, note that if the nodes do not wait before starting the transmission ($\tau_v = 0 : \ \forall v \in V$), then we will have $\boldsymbol{U}_T = \boldsymbol{I}_{\xi T \times \xi T}$.

6.C Proof of Theorem 6.1

For simplicity, in the following proof, we assume that each edge of the network has capacity 1. Edges with capacity more than 1 can be modeled by replacing them with multiple edges of unit capacity.

From (6.41) the transfer matrix from S to v at time T is equal to $\boldsymbol{H}^{(v)}[T]$. Knowing that the min-cut of node v is c_v, we choose a set of c_v incoming edges to v such that there exist c_v edge disjoint paths from S to v and find the input-output transfer matrix just for this set of edges. Then we can write

$$\begin{aligned} \hat{\boldsymbol{H}}^{(v)}[T] &= \hat{\boldsymbol{B}}_T^{(v)} (\boldsymbol{I} - \boldsymbol{U}_T \boldsymbol{F}_T)^{-1} \boldsymbol{U}_T \boldsymbol{A}_T \boldsymbol{M}_T \\ &= \hat{\boldsymbol{B}}_T^{(v)} \left(\boldsymbol{I} + \cdots + (\boldsymbol{U}_T \boldsymbol{F}_T)^{T-1} \right) \boldsymbol{U}_T \boldsymbol{A}_T \boldsymbol{M}_T, \end{aligned} \tag{6.43}$$

where $\hat{\boldsymbol{H}}^{(v)}[T] \in \mathbb{F}_q^{c_v \times n}$ and $\hat{\boldsymbol{B}}_T^{(v)} \in \mathbb{F}_q^{c_v \times \xi T}$. Let $f_{ij}^{(t,k)}$ denote for the entries of $\boldsymbol{F}_k^{(t)}$ and $m_{ij}^{(t)}$ denote for the entries of $\boldsymbol{M}[t]$. Every node in the network performs random linear NC so $m_{ij}^{(t)}$ and $f_{ij}^{(t,k)}$ (those that are not zero) are chosen uniformly at random from $\mathbb{F}_q$.

From (6.43) we know that each entry of $\hat{\boldsymbol{H}}^{(v)}[T]$ is a polynomial of degree at most T in variables $m_{ij}^{(t)}$ and $f_{ij}^{(t,k)}$. For $T > t_0(v)$ where $t_0(v) \triangleq \max_{i\in P(v)} \tau_i$, we know that there exists a trivial solution for variables $m_{ij}^{(t)}$ and $f_{ij}^{(t,k)}$ (which simply routes c_v packets from S to v through the c_v edge disjoint paths) that results in

$$\hat{\boldsymbol{H}}^{(v)}[T] = \left[\begin{array}{cc} \boldsymbol{I}_{c_v} & \boldsymbol{0}_{c_v\times(n-c_v)} \end{array} \right]. \tag{6.44}$$

Note that by changing the routing solution (in fact by changing the variables $m_{ij}^{(t)}$ properly) we could change the place of identity matrix in (6.44) arbitrarily. We conclude that the determinant of every $c_v \times c_v$ sub-matrix of $\hat{\boldsymbol{H}}^{(v)}[T]$ (which is a polynomial of degree at most c_vT in variables $m_{ij}^{(t)}$ and $f_{ij}^{(t,k)}$) is not identical to zero. So by using the Schwartz-Zippel lemma [82] we can upper bound the probability that $\hat{\boldsymbol{H}}^{(v)}[T]$ is not full rank if the variables $m_{ij}^{(t)}$ and $f_{ij}^{(t,k)}$ are chosen uniformly at random as follows

$$\mathbb{P}\left[\operatorname{rank} \hat{\boldsymbol{H}}^{(v)}[T] < c_v\right] < \frac{c_vT}{q}. \tag{6.45}$$

We can apply the same argument for $k < \frac{n}{c_v}$ consecutive time-slots to show that

$$\mathbb{P}\left[\operatorname{rank} \hat{\boldsymbol{H}}^{(v)}[T:T+k-1] < kc_v\right] < \frac{kc_v(T+k)}{q}, \tag{6.46}$$

where

$$\hat{\boldsymbol{H}}^{(v)}[T:T+k-1] \triangleq \left[\begin{array}{c} \hat{\boldsymbol{H}}^{(v)}[T] \\ \vdots \\ \hat{\boldsymbol{H}}^{(v)}[T+k-1] \end{array} \right]. \tag{6.47}$$

Now let us define the event $\mathcal{A}_k(v)$ as follows

$$\mathcal{A}_k(v):\ \operatorname{rank} \hat{\boldsymbol{H}}^{(v)}[T:T+k-1] = kc_v. \tag{6.48}$$

Then we can write

$$\begin{aligned} \mathbb{P}\left[\cap_{v\in V}\mathcal{A}_k(v)\right] &= 1 - \mathbb{P}\left[\cup_{v\in V}\mathcal{A}_k^{\mathfrak{C}}(v)\right] \\ &\geq 1 - \sum_{v\in V}\mathbb{P}\left[\mathcal{A}_k^{\mathfrak{C}}(v)\right] \\ &\geq 1 - \frac{k(T+k)}{q}\sum_{v\in V} c_v, \end{aligned} \tag{6.49}$$

where $T > t_0$ and $t_0 \triangleq \max_{v\in V} t_0(v)$.

This means that assuming q is large enough we are sure that with high probability each node v receives c_v innovative packets per time slot for $t > t_0$.

Part III

Secrecy

Overview

Secure communication is an important requirement for any communication system. In this chapter, we consider the problem of secret sharing among multiple nodes in a network (wireless or wired) in the presence of a passive eavesdropper. We consider a situation where there is a broadcast channel from one of the trusted nodes to the rest of them including the eavesdropper. Moreover, we assume that the legitimate nodes can discuss over a rate-unlimited public channel. Although in the literature there have been proposed upper and lower bounds for the secret key sharing capacity for this scenario (e.g., see [83, 84, 85, 86]), the exact characterization for the secrecy capacity is still an open problem.

In this part, we focus on some specific cases of the aforementioned problem and our goal is to propose secret key sharing schemes that perform well and are efficient. To be more specific, we start Chapter 7 by studying the erasure broadcast channel. The proposed achievability scheme for this problem achieves the secrecy capacity and unlike many information theoretical problems it is practically efficient (i.e., it is a polynomial time algorithm). Then we extend the two-state erasure channel model to a multi-state deterministic channel model. For this problem the secrecy capacity is characterized as well and it is achieved by an efficient scheme. The multi-state deterministic channel is a model to capture the different SNR level in a wireless network. By considering this problem, we obtain some insights about the state-dependent Gaussian model which is the last scenario we consider in Chapter 7. By applying a nested message set, degraded channel wiretap code, and by using the insights we gain by studying the deterministic broadcast channel, we may convert the state-dependent Gaussian broadcast channel to multiple parallel and independent erasure channels. The final achievability scheme for this problem is not optimal in general. However, for the high dynamic range case when SNR is high, we show that it achieves the optimal performance.

In Chapter 8, we consider a similar problem as introduced above but instead over a non-coherent NC multicast channel which is modeled by a multiplicative matrix channel with random uniform distribution over channel transfer matrices, as introduced in Chapter 3 (see Section 3.1.2). By using the insight we gain from the study of the erasure broadcast channel in Chapter 7, we propose an

efficient achievability scheme for this problem. The achievable rate is given by the solution of a convex optimization problem. However, because of the fundamental differences that subsets and subspaces have, the final achievable secret key rate is not optimum in general.

"In all secrets there is a kind of guilt, however beautiful or joyful they may be, or for what good end they may be set to serve. Secrecy means evasion, and evasion means a problem to the moral mind."

- Gilbert Parker

Group Secret Key Agreement over Wireless Broadcast Channels

7

In this chapter we consider the problem of generating a secret key $\mathcal{K}$ among $m+1 \geq 2$ honest nodes that communicate over a wireless channel in the presence of a passive eavesdropper, Eve. We restrict our attention to the case where communication occurs either through a broadcast channel, where the received symbols are independent among all receivers of the broadcast transmissions (including Eve) given that the transmitted symbols is known, or, through a no-cost public channel.

Here, we consider three types of broadcast channels. First, we model a wireless broadcast channel by a *packets erasure channel*. For this model we characterize the *secret key generation* capacity and propose a computationally efficient achievability scheme that employs techniques from NC. Surprisingly, we show that the rates at which we can generate a secret key among the $m+1$ nodes, is not affected by the number of nodes; that is, whether we try to establish a secret key between two nodes, or an arbitrary number, we can do this at the same rate. This result is reminiscent of the main theorem in NC (see Theorem 2.1), where a source can multicast information to a set of receivers at the same rate, independently of the number of receivers [6].

However, in wireless communication systems when a packet is declared to be erased, it does not mean that the whole packet symbols are corrupted. Depending on the SNR [1] level, we might have more or less number of corrupted symbols. One way to model this behavior is to consider a multi-state channel in contrast to the two-state erasure channel.

Thus, we continue the chapter by initiating the study of group secret key agreement over a *state-dependent Gaussian broadcast channel*. This can be motivated by fading wireless channels, where the channel states vary over time. The use of state-dependent channels for secrecy has been of interest recently

1. Signal to noise ratio.

(see for example [87] and references therein). To gain insight into our problem, we first investigate a *deterministic approximation* of the wireless channel as defined in [88, 89]. For the deterministic broadcast channel we will show that using a superposition based secrecy scheme [90], we can develop a group key agreement protocol that can be shown to be information-theoretically optimal. This can be done by converting the deterministic channel to multiple independent erasure channels. In particular, we show that we can can get the same key agreement rate for the entire group as we would get for a single pair of nodes. Therefore this result demonstrates that with unlimited public discussion, we get secret key-agreement rates for linear deterministic channels, that is invariant to network size. Similar to the case of erasure broadcast channel, a key idea to get this is a connection to NC, which allows efficient reconciliation of the group secret.

We use the deterministic achievability scheme to get an insight about the Gaussian wireless broadcast channel with state. To this end, we use a nested message set, degraded channel wiretap code based on the broadcast approach of [90] to develop a key-agreement protocol for the noisy broadcast problem. This enables a scheme that converts the wireless channel with state to behave similar to the deterministic case. Though this is not optimal, we can demonstrate that when there is a large dynamic range between the channel states, this scheme is optimal in the (generalized) degrees of freedom sense.

It is important to mention that Section 7.6 and Section 7.7 have been done as a joint work with Shaunak Mishra [2].

7.1 Related Work

Secret key generation over wireless channels is a problem that has attracted significant interest. In a seminal paper on "wiretap" channels, Wyner [91] pioneered the notion that one can establish information-theoretic secrecy between Alice and Bob by utilizing the noisy broadcast nature of wireless transmissions. However, his scheme works only if we have perfect knowledge of Eve's channel and moreover, only if Eve has a worse channel than Bob. In a subsequent seminal work, Maurer [84] showed the value of feedback from Bob to Alice, even if Eve hears all the feedback transmissions (i.e., the feedback channel is public). He showed that even if the channel from Alice to Eve is better than that to Bob, feedback allows Alice and Bob to create a key which is information-theoretically secure from Eve. The problem of key agreement between a set of terminals with access to noisy broadcast channel and public discussion channel (visible to the eavesdropper) was studied in [85], where some achievable secrecy rates were established, assuming Eve does not have access to the noisy broadcast transmissions. The case when the eavesdropper also had access to the broadcast channel was the main focus of recent work in [92, 86] which developed (non-computable) lower and upper bounds for secrecy rates.

2. Shaunak Mishra is a Ph.D. student at University of California, Los Angeles (UCLA), working under the supervision of prof. Suhas Diggavi.

To the best of our knowledge, ours is the first work to consider multi-terminal secret key agreement over erasure networks and wireless broadcast channels with state, when Eve also has access to the noisy broadcast transmissions. Moreover, unlike the information-theoretic works in [91, 84, 83, 85, 86] that assume infinite complexity operations, our schemes for the erasure and deterministic broadcast channels are computationally efficient.

7.2 Problem Statement

We consider a set of $m+1 \geq 2$ honest nodes, $\mathsf{T}_0, \ldots, \mathsf{T}_m$, that [3] aim to share a secret key $\mathcal{K}$ among themselves while keeping it concealed from a passive adversary, Eve. Eve does not perform any transmissions, but is trying to eavesdrop on (overhear) the communications between the honest nodes. For convenience, sometimes we will refer to legitimate terminals $\mathsf{T}_0, \mathsf{T}_1, \mathsf{T}_2, \ldots$, as "Alice," "Bob," "Calvin," and so on [4].

We assume that Alice has access to a broadcast channel such that the rest of the terminals (including Eve) receive independent noisy version of what she broadcasts, i.e., for the channel transition probability we can write

$$P_{X_1 \cdots X_m X_{\mathsf{E}}|X_{\mathsf{A}}}(x_1, \ldots, x_m, x_{\mathsf{E}}|x_{\mathsf{A}}) = P_{X_{\mathsf{E}}|X_{\mathsf{A}}}(x_{\mathsf{E}}|x_{\mathsf{A}}) \prod_{i=1}^{m} P_{X_i|X_{\mathsf{A}}}(x_i|x_{\mathsf{A}}), \quad (7.1)$$

where the input and output symbols of the channel are from some arbitrary sets. We also assume that all of the honest terminals can discuss over a cost-free public channel where everybody (including Eve) can hear the discussion.

In the following, we define a protocol that abstracts the interactive communication between terminals aiming to share a common secret key $\mathcal{K}$ (see also [84, 83, 85, 86]).

Definition 7.1. *The* secret key generating protocol *is defined as follows:*

1. *For $t = 0$, all of the honest terminals generate independent random variables $W_0, \ldots, W_m$.*
2. *(i) For time $1 \leq t \leq n$, Alice transmits $X_0[t]$ over the broadcast channel where*

$$X_0[t] = X_{0,t}\left(W_0, \boldsymbol{\mathcal{D}}^{t-1}\right). \quad (7.2)$$

We will define the random vectors $\boldsymbol{\mathcal{D}}[t]$ in the following. Then the other terminals receive $X_1[t], \ldots, X_m[t]$ and Eve receives $X_{\mathsf{E}}[t]$.
(ii) Following each of the broadcast transmissions, there is the possibility for the terminals to discuss over a cost-free public channel. This discussion continues for $r[t]$ rounds and is represented by the random vectors

3. We use T to denote for "terminal".

4. During this chapter, we use T_i and i interchangeably when they are used as subscript. So, for example, instead of X_{T_i} we sometimes write X_i. At some points, we also use X_{A}, X_{B}, and etc. to denote for X_0 and X_1, and so on.

$\boldsymbol{\mathcal{D}}[t] = \big(\mathcal{D}_0[t], \ldots, \mathcal{D}_{r[t]}[t]\big)$, *where*

$$\mathcal{D}_i[t] = \mathcal{D}_{i,t}\left(W_j, X_j^t, \boldsymbol{\mathcal{D}}^{t-1}, \mathcal{D}_{0:i-1}[t]\right) \tag{7.3}$$

is the public message revealed by the jth terminal with $j = i \mod (m+1)$ (in other words, the indexing of the discussion is done in a round robin order).

3. *Finally, the ith terminal creates a key $\mathcal{K}_i$ where*

$$\mathcal{K}_i = \mathcal{K}_i(W_i, X_i^n, \boldsymbol{\mathcal{D}}^n). \tag{7.4}$$

Definition 7.2. *A number R_s is called an achievable key generation rate if for every $\epsilon > 0$ and sufficiently large n there exists a key generating protocol as defined in Definition 7.1 such that we have*

$$\mathbb{P}\left[\mathcal{K}_i \neq \mathcal{K}_j\right] < \epsilon, \quad \forall i, j : i \neq j, \tag{7.5}$$

$$I(\mathcal{K}_0; X_{\mathsf{E}}^n, \boldsymbol{\mathcal{D}}^n) < \epsilon, \tag{7.6}$$

and

$$\frac{1}{n} H(\mathcal{K}_0) > \mathfrak{R}_s - \epsilon. \tag{7.7}$$

The supremum of the achievable key rate as $n \to \infty$ and $\epsilon \to 0$ is called the key generation capacity C_s.

In the following, we will be going to introduce the different broadcast channels we will use in this chapter to model a wireless network.

7.2.1 Erasure Broadcast Channels

Here, we introduce the erasure broadcast channel model. We assume the input and output symbols to the erasure channel are packets of length L of elements from a finite filed $\mathbb{F}_q$. When Alice transmits a packet, node T_i correctly overhears it with probability $1 - \delta_i$, where δ_i is the "erasure probability" of the channel. Similarly, Eve correctly receives the packets with probability $1 - \delta_{\mathsf{E}}$. The erasure events happen independently over time and across different channels. For simplicity (and without loss of generality) in this chapter we will focus on the symmetric case where we have $\delta_i = \delta$.

7.2.2 Deterministic Broadcast Channels

In this section, we introduce the deterministic model for the Gaussian channels which will be introduced in Section 7.2.3.

Let the transmitted vector sent by Alice be denoted by $\boldsymbol{X}_{\mathsf{A}} \in \mathbb{F}_q^L$. The received vectors for every terminal and Eve depend on their channel states for the particular time instant. We define a random variable $S_{\mathsf{T}_i} \in [0 : s]$ corresponding to the state of the channel for the ith terminal and similarly

define the random variable $S_{\mathsf{E}} \in [0:s]$ for Eve. We assume that the channel states are independent and for each receiver $r \in \{\mathsf{T}_1, \ldots, \mathsf{T}_m, \mathsf{E}\}$ we have [5]

$$\mathbb{P}[S_r = k] = \delta_k, \qquad k \in [0:s]. \tag{7.8}$$

Then we model the received vector at the receiver r by a state-dependent deterministic channel as follows

$$\widetilde{\boldsymbol{X}}_r[t] = \boldsymbol{F}_{S_r[t]} \boldsymbol{X}_{\mathsf{A}}[t], \qquad \forall r \in \{\mathsf{T}_1, \ldots, \mathsf{T}_m, \mathsf{E}\}, \tag{7.9}$$

where $\boldsymbol{F}_i \in \mathbb{F}_q^{L \times L}$ for $i \in [0:s]$. Moreover, we assume that the state of a particular channel is available at the corresponding receiver. We define a composite received vector for the receiver r as follows

$$\boldsymbol{X}_r[t] = (\widetilde{\boldsymbol{X}}_r[t], S_r[t]). \tag{7.10}$$

In order to capture and model the different SNR level for the Gaussian channel we use the shift matrix model developed in [88, 89]. This implies that the matrices $\boldsymbol{F}_i$ have the following nested structure:

$$\boldsymbol{0} = \ker \boldsymbol{F}_s \subset \ker \boldsymbol{F}_{s-1} \subset \cdots \subset \ker \boldsymbol{F}_0 = \mathbb{F}_q^L, \tag{7.11}$$

and

$$\operatorname{rank}(\boldsymbol{F}_i - \boldsymbol{F}_{i-1}) = \operatorname{rank}(\boldsymbol{F}_i) - \operatorname{rank}(\boldsymbol{F}_{i-1}). \tag{7.12}$$

For convenience we assume that $\boldsymbol{F}_s = \boldsymbol{I}_L$. The two extreme states "0" and "s" correspond to complete erasure and complete receiving of the transmitted vector $\boldsymbol{X}_{\mathsf{A}}$. This deterministic model is indeed an extension to the packet erasure broadcast channel, introduced in Section 7.2.1, which only has two channel states.

7.2.3 State-Dependent Gaussian Broadcast Channels

Finally, in this section, we introduce the state-dependent additive white Gaussian broadcast channel model where for each receiver the channel state remains the same for a block of symbols of length L and changes independently from one block to another block. We also assume that L is large enough that enables us to apply information theoretical arguments within each block. The transmitted vector sent by Alice is denoted by $\boldsymbol{X}_{\mathsf{A}} \in \mathbb{R}^L$. The received vectors for every terminal and Eve depend on their channel states for the particular time instant. We define a random variable $S_{\mathsf{T}_i} \in [0:s]$ corresponding to the state of the channel for the ith terminal and similarly define the random variable $S_{\mathsf{E}} \in [0:s]$ for Eve. For the channel state of a receiver $r \in \{\mathsf{T}_1, \ldots, \mathsf{T}_m, \mathsf{E}\}$ we assume that [6]

$$\mathbb{P}[S_r = k] = \delta_k, \qquad k \in [0:s]. \tag{7.13}$$

5. For simplicity of demonstration and without loss of generality, here we only consider a symmetric case where the probability distribution over the channel states are the same for all of the receivers.

6. Again, for simplicity, we consider a symmetric problem where the probability distribution over the states are the same for all of the receivers (including Eve). Moreover, we focus on a finite number of states. Both these restrictions can be relaxed.

Then we model the received vector at the receiver r by a state-dependent white Gaussian channel as follows

$$\widetilde{\boldsymbol{X}}_r[t] = h_{S_r[t]}\boldsymbol{X}_{\mathsf{A}}[t] + \boldsymbol{Z}_r[t], \qquad \forall r \in \{\mathsf{T}_1, \ldots, \mathsf{T}_m, \mathsf{E}\} \tag{7.14}$$

where $\widetilde{\boldsymbol{X}}_r[t] \in \mathbb{R}^L$ and $\boldsymbol{Z}_r[t] \in \mathbb{R}^L$. For the additive noise of each receiver we have $\boldsymbol{Z}_r[t] \sim \mathcal{N}(0, \boldsymbol{I}_L)$. The channel gains h_i are some real constants such that

$$h_0 \leq \cdots \leq h_s. \tag{7.15}$$

We also assume that the channel input is subject to an average power constraint P, i.e.,

$$\frac{1}{L}\mathbb{E}\left[\|\boldsymbol{X}_{\mathsf{A}}\|^2\right] \leq P. \tag{7.16}$$

Moreover, we assume that the channel state information (CSI) is completely known by each receiver. So we define a composite received vector for each receiver r as follows

$$\boldsymbol{X}_r[t] = (\widetilde{\boldsymbol{X}}_r[t], S_r[t]). \tag{7.17}$$

7.3 Main Results

The main results of this chapter is summarized in the following.

Theorem 7.1. *The secret key generation capacity among $m+1$ terminals, as defined in Section 7.2, that have access to an erasure broadcast channel (see Section 7.2.1) is given by*

$$C_s^{\mathsf{ers}} = (1-\delta)\delta_{\mathsf{E}}\left(L \log q\right), \tag{7.18}$$

where δ is the erasure probability from Alice to the rest of terminals and δ_{E} is the erasure probability from Alice to Eve.

The converse part of Theorem 7.1 is stated in Section 7.5.1 (Theorem 7.6) and the achievability part is stated in Section 7.5.2 (Theorem 7.7).

Theorem 7.2. *The secret key generation capacity among $m+1$ terminals that have access to a state-dependent deterministic broadcast channel (see Section 7.2.2) is given by*

$$C_s^{\mathsf{det}} = \sum_{j=1}^{s} \left[\operatorname{rank} \boldsymbol{F}_j - \operatorname{rank} \boldsymbol{F}_{j-1}\right] \left(\sum_{i=0}^{j-1} \rho_i\right) \log q, \tag{7.19}$$

where $\rho_i \triangleq \delta_i - 2\delta_i(\delta_0 + \cdots + \delta_{i-1})\mathbb{1}_{\{i>0\}} - \delta_i^2$.

The proof of Theorem 7.2 is stated in Section 7.6.

Theorem 7.3. *The secret key generation capacity among $m+1$ terminals that have access to a state-dependent Gaussian broadcast channel (see Section 7.2.3) is upper bounded by*

$$C_s^{\mathsf{gaus}} \leq \frac{1}{2} L \sum_{i=0}^{s} \sum_{j=0}^{s} \delta_i \delta_j \log \left(1 + \frac{h_i^2 P}{1 + h_j^2 P} \right). \tag{7.20}$$

Also the secrecy capacity can be lower bounded by the solution of the following optimization problem

$$C_s^{\mathsf{gaus}} \geq \left\{ \begin{array}{ll} \max & \sum_{i=1}^{s} \Delta_i L R_i \\ \text{subject to} & \sum_{i=1}^{s} P_i = P \\ & P_i \geq 0, \quad \forall i \in [1:s], \end{array} \right. \tag{7.21}$$

where R_i is given in (7.75) and $\Delta_i \triangleq (1-\theta_i)\theta_i$ such that $\theta_i = \sum_{j=0}^{i-1} \delta_j$.

Moreover, for the high dynamic range case where $h_i \gg h_{i-1}, \forall i \in [1:s]$, and when we are in high SNR regime we can write

$$C_s^{\mathsf{gaus}} \doteq \frac{1}{2} L \sum_{i=1}^{s} \Delta_i \log \frac{h_i^2}{h_{i-1}^2}, \tag{7.22}$$

where we use "$\doteq$" for the exponential equality.

7.4 Upper Bound for the Key Generation Capacity of Independent Broadcast Channels

The secret key generation capacity among multiple terminals (without eavesdropper having access to the broadcast channel) is completely characterized in [85]. By using this result, it is possible to state an upper bound for the secrecy capacity of the key generation problem among multiple terminals where the eavesdropper has also access to the broadcast channel. This can be done by adding a dummy terminal to the first problem and giving all the eavesdropper's information to this dummy node and let it to participate in the key generation protocol. By doing so, the secret key generation rate does not decrease. Hence by combining [85, Theorem 4.1] and [85, Lemma 5.1], the following result can be stated.

Theorem 7.4. *The secret key generation capacity among $m+1$ terminals as defined in Definition 7.2, is upper bounded as follows*

$$C_s \leq \max_{P_{X_0}} \min_{\lambda \in \Lambda([0:m])} \left[H(X_{[0:m]}|X_{\mathsf{E}}) - \sum_{B \subsetneq [0:m]} \lambda_B H(X_B|X_{B^c}, X_{\mathsf{E}}) \right], \tag{7.23}$$

where $\Lambda([0:m])$ is the set of all collections $\lambda = \{\lambda_B : B \subsetneq [0:m], B \neq \emptyset\}$ of weights $0 \leq \lambda_B \leq 1$, satisfying

$$\sum_{B \subsetneq [0:m], i \in B} \lambda_B = 1, \qquad \forall i \in [0:m]. \tag{7.24}$$

Note that in the above expression for the upper bound, it is possible to change the order of maximization and minimization [85, Theorem 4.1].

Now, for our problem where the channel from Alice to the other terminals are assumed to be independent, we can further simplify the upper bound given in Theorem 7.4, as stated in Theorem 7.5.

Theorem 7.5. *If the channels from Alice to the other terminals are independent, as described in* (7.1)*, then the upper bound stated in Theorem 7.4, for the secret key generation capacity is simplified to*

$$C_s \leq \max_{P_{X_0}} \min_{j\in[1:m]} I(X_0; X_j|X_{\mathsf{E}}) \tag{7.25}$$

$$\leq \min_{j\in[1:m]} \max_{P_{X_0}} I(X_0; X_j|X_{\mathsf{E}}). \tag{7.26}$$

Proof. First, note that we can write

$$\begin{aligned} H(X_{[0:m]}|X_{\mathsf{E}}) &\overset{\text{(a)}}{=} \sum_{j=0}^{m} H(X_i|X_{\mathsf{E}}, X_{[0:j-1]}) \\ &\overset{\text{(b)}}{=} H(X_0|X_{\mathsf{E}}) + \sum_{j=1}^{m} H(X_j|X_0), \end{aligned} \tag{7.27}$$

where (a) follows from the chain rule and (b) follows from the independence of the channels. Similarly, for every $B \subsetneq [0:m]$ we can expand $H(X_B|X_{B^c}, X_{\mathsf{E}})$ as follows

$$H(X_B|X_{B^c}, X_{\mathsf{E}}) = H(X_0|X_{B^c}, X_{\mathsf{E}}) + \sum_{j\in B} H(X_j|X_0). \tag{7.28}$$

Now, from Theorem 7.4, we know that for every $\lambda \in \Lambda([0:m])$ there exists a distribution P_{X_0} such that C_s is upper bounded by

$$\begin{aligned} C_s &\leq H(X_{[0:m]}|X_{\mathsf{E}}) - \sum_{B\subsetneq[0:m]} \lambda_B H(X_B|X_{B^c}, X_{\mathsf{E}}) \\ &= H(X_0|X_{\mathsf{E}}) + \sum_{j=1}^{m} H(X_j|X_0) \\ &\quad - \sum_{B\subsetneq[0:m]} \lambda_B \left[H(X_0|X_{B^c}, X_{\mathsf{E}}) + \sum_{j\in B} H(X_j|X_0) \right] \\ &\overset{\text{(a)}}{=} H(X_0|X_{\mathsf{E}}) - \sum_{B\subsetneq[0:m],\ 0\in B} \lambda_B H(X_0|X_{B^c}, X_{\mathsf{E}}) \\ &\quad + \sum_{j=1}^{m} H(X_j|X_0) - \sum_{j=1}^{m} \sum_{B\subsetneq[0:m],\ j\in B} \lambda_B H(X_j|X_0) \end{aligned}$$

$$\stackrel{(b)}{=} H(X_0|X_{\mathsf{E}}) - \sum_{B\subsetneq[0:m],\ 0\in B} \lambda_B H(X_0|X_{B^c}, X_{\mathsf{E}})$$
$$= \sum_{B\subsetneq[0:m],\ 0\in B} \lambda_B I(X_0; X_{B^c}|X_{\mathsf{E}}), \tag{7.29}$$

where in (a) we have changed the order of summation over j and B, and (b) follows from (7.24). In order to find the best upper bound we proceed as follows. For every λ and P_{X_0} we can write

$$\begin{aligned} A &\triangleq \sum_{B\subsetneq[0:m],\ 0\in B} \lambda_B I(X_0; X_{B^c}|X_{\mathsf{E}}) \\ &\geq \sum_{B\subsetneq[0:m],\ 0\in B} \lambda_B \min_{j\in B^c} I(X_0; X_j|X_{\mathsf{E}}) \\ &\geq \sum_{B\subsetneq[0:m],\ 0\in B} \lambda_B \min_{j\in[1:m]} I(X_0; X_j|X_{\mathsf{E}}) \\ &= \min_{j\in[1:m]} I(X_0; X_j|X_{\mathsf{E}}). \end{aligned} \tag{7.30}$$

Let us define $i = \arg\min_{j\in[1:m]} I(X_0; X_j|X_{\mathsf{E}})$. Then, note that $\lambda_B = \lambda_{B^c} = 1$ where $B^c = \{i\}$ is a valid choice according to the condition of Theorem 7.4, i.e., they satisfy (7.24). Now, for this choice we have the chain of inequalities in (7.30) is satisfied with equalities.

Combining all of the above arguments, for the secrecy upper bound we can write

$$\begin{aligned} C_s &\leq \max_{P_{X_0}} \min_{j\in[1:m]} I(X_0; X_j|X_{\mathsf{E}}) \\ &\leq \min_{j\in[1:m]} \max_{P_{X_0}} I(X_0; X_j|X_{\mathsf{E}}). \end{aligned} \tag{7.31}$$

□

Remark 7.1. *Note that* (7.25) *is the best upper bound one might hope for an independent broadcast channel using the results of [85].*

Remark 7.2. *Using [84, Theorem 7] or [83, Theorem 2], we observe that the bound given in* (7.26) *is indeed tight for the two terminals problem where we have the Markov chains* $X_{\mathsf{B}} \leftrightarrow X_{\mathsf{A}} \leftrightarrow X_{\mathsf{E}}$ *(when the channels are independent) or* $X_{\mathsf{A}} \leftrightarrow X_{\mathsf{B}} \leftrightarrow X_{\mathsf{E}}$ *(when the channels are degraded).*

7.5 Group Secret Key Agreement over Erasure Broadcast Channels

In this section, we will be going to characterize the secret key generation capacity among multiple terminals communicating over an erasure broadcast channel and state a proof for Theorem 7.1.

7.5.1 Upper Bound for the Key Generation Capacity

From Theorem 7.5, we can state an upper bound for the secret key sharing capacity among multiple terminals where Alice broadcasts information over an erasure channel as given in the following result, Theorem 7.6.

Theorem 7.6. *The key generation capacity defined in Definition 7.2 can be upper bounded as follows*

$$C_s^{\mathsf{ers}} \leq (1-\delta)\delta_{\mathsf{E}}\left(L\log q\right). \tag{7.32}$$

Proof. As mentioned before, by using Theorem 7.5 we can upper bound the secrecy capacity as follows

$$\begin{aligned} C_s^{\mathsf{ers}} &\leq \min_{j\in[1:m]} \max_{P_{\boldsymbol{X}_{\mathsf{A}}}} I(\boldsymbol{X}_{\mathsf{A}}, \boldsymbol{X}_j|\boldsymbol{X}_{\mathsf{E}}) \\ &\overset{(a)}{=} \max_{P_{\boldsymbol{X}_{\mathsf{A}}}} I(\boldsymbol{X}_{\mathsf{A}}, \boldsymbol{X}_{\mathsf{B}}|\boldsymbol{X}_{\mathsf{E}}) \\ &= \max_{P_{\boldsymbol{X}_{\mathsf{A}}}} \left[H(\boldsymbol{X}_{\mathsf{A}}|\boldsymbol{X}_{\mathsf{E}}) - H(\boldsymbol{X}_{\mathsf{A}}|\boldsymbol{X}_{\mathsf{E}}, \boldsymbol{X}_{\mathsf{B}})\right] \\ &= \max_{P_{\boldsymbol{X}_{\mathsf{A}}}} \left[\delta_{\mathsf{E}} - \delta_{\mathsf{E}}\delta\right] H(\boldsymbol{X}_{\mathsf{A}}) \\ &= (1-\delta)\delta_{\mathsf{E}}\left(L\log q\right), \end{aligned} \tag{7.33}$$

where (a) follows because of the symmetry of the problem. Note that $\boldsymbol{X}_{\mathsf{B}}$ and $\boldsymbol{X}_{\mathsf{E}}$ are random variables which are "erased" versions of $\boldsymbol{X}_{\mathsf{A}}$, with erasure probabilities δ and δ_{E} respectively. This concludes the theorem. □

7.5.2 Lower Bound for the Key Generation Capacity

Here we describe and analyze our achievability scheme for the secret key generation over erasure broadcast channels. The proposed scheme consists of several phases and it proceeds as follows.

Private Phase:

1. Alice broadcasts n packets, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, where $\boldsymbol{x}_i \in \mathbb{F}_q^L$ and $\boldsymbol{x}_i \sim \mathsf{Uni}\left(\mathbb{F}_q^L\right)$ (we will call them "x-packets"). Of these, n^* packets are received by at least one honest node. This set is denoted by N^* where $n^* = |N^*|$.

Public Discussion (Initial Phase):

1. Each honest node sends Alice publicly a feedback message specifying which x-packets it received. Let I_{T_i} denotes the set of packets' indices received by the ith terminal T_i.
2. Alice constructs $h = \delta_{\mathsf{E}} \cdot n^*$ linear combinations of the x-packets, $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_h$ (we will call them "y-packets"), as follows:

   (i) She divides the set N^* of x-packets that were received by at least one honest node into non-overlapping subsets, such that each subset consists of all the packets that were commonly received by a different subset of

honest nodes. To be more precise, let S be an arbitrary non-empty subset of $[1:m]$ and let us define the set

$$N_{S,\overline{S}} \triangleq \big\{ i \in [1:n] \mid i \in I_{\mathsf{T}_j} : \forall j \in S, \text{ and } i \notin I_{\mathsf{T}_j} : \forall j \notin S \big\}. \tag{7.34}$$

Then we have

$$N^* = \bigcup_{\emptyset \neq S \subseteq [1:m]} N_{S,\overline{S}}. \tag{7.35}$$

(ii) From each such subset of packets $N_{S,\overline{S}}$, she creates $\delta_{\mathsf{E}} \cdot n_{S,\overline{S}}$ linear combinations using the construction described in Lemma 7.1 (provided in the Appendix 7.A), where $n_{S,\overline{S}} \triangleq |N_{S,\overline{S}}|$.

Then she publicly reveals the coefficients she used to create all the y-packets.

3. Each node T_i reconstructs as many (say h_i) of the y-packets as it can (based on the x-packets it received in step #1). For h_i we can write

$$h_i = \sum_{\emptyset \neq S \subseteq [1:m]:\ i \in S} \delta_{\mathsf{E}} \cdot n_{S,\overline{S}}. \tag{7.36}$$

Note that as n grows we have $h_i \to \mathbb{E}[h_i]$ which is equal to

$$\mathbb{E}[h_i] = (1-\delta)\delta_{\mathsf{E}} n. \tag{7.37}$$

Public Discussion (Reconciliation Phase):

1. Alice creates $h - \min_i h_i$ linear combinations of the y-packets (we will call them "z-packets"), using the construction provided in Lemma 7.2 (provided in the Appendix 7.A). She publicly reveals both the contents and the coefficients of the z-packets, such that each node T_i receives at least $h - h_i$ of them.
2. Each node T_i combines the $h - h_i$ z-packets it received with the h_i y-packets it recreated in phase 1, and reconstructs all the y-packets.
3. Alice creates $l = \min_i h_i$ linear combinations of the y-packets, $k_1, \ldots, k_l$ (we will call them "k-packets"), using the construction stated in Lemma 7.3 (provided in the Appendix 7.A). She publicly reveals the coefficients she used to create all the k-packets.
4. Each node T_i reconstructs all the k-packets. The common secret key is the concatenation of all the k-packets, $\mathcal{K} = \{k_1, \ldots, k_l\}$.

Now we may summarize the above achievability scheme as follows based on Definition 7.1. At $t = 0$ Alice generates the random variable $W_0 = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ where $\boldsymbol{x}_i \sim \mathsf{Uni}\left(\mathbb{F}_q^L\right)$. We have also $W_{1:m} = \emptyset$. For each time t, $1 \leq t < n$, she broadcasts $\boldsymbol{X}_0[t] = \boldsymbol{x}_t$ and there is no public discussions afterwards; namely we have $\boldsymbol{\mathcal{D}}[t] = \emptyset$. After the nth transmission by Alice there is a public discussion in many rounds; simplified as follows. We have $\boldsymbol{\mathcal{D}}[n] = (P_1, P_2, P_3, P_4)$ where P_1 denotes the set of indices I_{T_i} that have been sent back by the honest terminals, P_2 denotes the coefficients of the y-packets, P_3 denotes the z-packets and their coefficients, and finally P_4 represents the coefficients of k-packets.

Theorem 7.7. *The achievable secret key generation rate of the above scheme is*

$$\mathfrak{R}_s^{\mathsf{ers}} = (1-\delta)\delta_{\mathsf{E}}\left(L\log q\right). \tag{7.38}$$

Proof. The way that the achievability scheme is proposed, constructively satisfies Condition (7.5).

To prove Condition (7.7) we proceed as follows. Let us define $\overline{l} \triangleq l/n$. Then we can write

$$\begin{aligned} H(\mathcal{K}) &= H(\mathcal{K},\overline{l}) = H(\mathcal{K}|\overline{l}) + H(\overline{l}) \\ &\geq H(\mathcal{K}|\overline{l}) \\ &= H(\mathcal{K}|\alpha < \overline{l} < \beta)\cdot \mathbb{P}\left[\alpha < \overline{l} < \beta\right] \\ &\quad + H(\mathcal{K}|\overline{l} \geq \beta)\cdot \mathbb{P}\left[\overline{l} \geq \beta\right] + H(\mathcal{K}|\overline{l} \leq \alpha)\cdot \mathbb{P}\left[\overline{l} \leq \alpha\right] \\ &\geq H(\mathcal{K}|\alpha < \overline{l} < \beta)\cdot \mathbb{P}\left[\alpha < \overline{l} < \beta\right] \\ &\geq n\alpha\left(L\log q\right)\left[1 - \mathbb{P}\left[\overline{l} \leq \alpha\right] - \mathbb{P}\left[\overline{l} \geq \beta\right]\right], \end{aligned} \tag{7.39}$$

where $\alpha = \mu - \gamma$ and $\beta = \mu + \gamma$ for some small γ, $0 < \gamma \leq \mu$, such that $\mu = (1-\delta)\delta_{\mathsf{E}}$. Then by applying the concentration result of Lemma 7.4 (see Appendix 7.A) we have

$$\mathbb{P}\left[\overline{l} \leq \alpha\right] \leq m\exp\left(-\frac{\gamma^2}{2\mu}n\right) \triangleq a, \tag{7.40}$$

and

$$\mathbb{P}\left[\overline{l} \geq \beta\right] \leq \exp\left(-\frac{m\gamma^2}{3\mu}n\right) \triangleq b. \tag{7.41}$$

Hence, we can observe that by choosing

$$\mathfrak{R}_s^{\mathsf{ers}} = \mu(L\log q) \tag{7.42}$$

and

$$\epsilon = \mu\left(L\log q\right)\left[a+b\right] + \gamma\left(L\log q\right)\left[1-a-b\right] \tag{7.43}$$

we have

$$\frac{1}{n}H(\mathcal{K}) > \mathfrak{R}_s^{\mathsf{ers}} - \epsilon \tag{7.44}$$

satisfied. Then, we get the desired result by making γ arbitrarily small because we have $\epsilon \to 0$ if $\gamma \to 0$.

To prove Condition (7.6) we need to show that

$$I(\mathcal{K}; \boldsymbol{X}_{\mathsf{E}}^n, P_1, P_2, P_3, P_4) < \epsilon. \tag{7.45}$$

By using a similar technique that we used above to bound $H(\mathcal{K})$, (using Lemma 7.1 and some concentration results for $n_{S,\overline{S}}$) we can show that

$$I(\boldsymbol{Y}; \boldsymbol{X}_{\mathsf{E}}^n, P_1, P_2) < \epsilon. \tag{7.46}$$

Using Lemma 7.3, by construction we have also

$$I(\mathcal{K}; P_3, P_4) = 0. \tag{7.47}$$

Now we know that the coefficients of the z-packets and k-packets form a basis (see Lemma 7.3) so the random variable $\boldsymbol{Y}$ and the random variable $(\mathcal{K}, P_3, P_4)$ are equivalent (having one we have the other). Then we can write (7.46) as follows

$$\begin{aligned} I(\boldsymbol{Y}; \boldsymbol{X}_{\mathsf{E}}^n, P_1, P_2) &= I(\mathcal{K}, P_3, P_4; \boldsymbol{X}_{\mathsf{E}}^n, P_1, P_2) \\ &= I(P_3, P_4; \boldsymbol{X}_{\mathsf{E}}^n, P_1, P_2) \\ &\quad + I(\mathcal{K}; \boldsymbol{X}_{\mathsf{E}}^n, P_1, P_2 | P_3, P_4) < \epsilon, \end{aligned} \tag{7.48}$$

so

$$I(\mathcal{K}; \boldsymbol{X}_{\mathsf{E}}^n, P_1, P_2 | P_3, P_4) < \epsilon. \tag{7.49}$$

Now we can expand

$$\begin{aligned} I(\mathcal{K}; \boldsymbol{X}_{\mathsf{E}}^n, P_1, P_2, P_3, P_4) &= I(\mathcal{K}; P_3, P_4) \\ &\quad + I(\mathcal{K}; \boldsymbol{X}_{\mathsf{E}}^n, P_1, P_2 | P_3, P_4), \end{aligned} \tag{7.50}$$

where the first term is zero by (7.47) and second term is very small because of (7.49), so we are done. □

Combining the results of Theorem 7.6 and Theorem 7.7 we have proved Theorem 7.1.

Remark 7.3. *It is worthwhile to mention that this result can be easily extended to the asymmetric case where the channels to the legitimate users are not statistically identical. Moreover, notice that in the symmetric case, the key-generation rate is the same for any $m \geq 1$, and therefore this protocol scales ideally with network size. Finally, the critical difference between $m = 1$ and $m > 1$ is that the key-reconciliation necessitated the use of ideas from NC, which we believe is a new observation.*

7.6 Group Secret Key Agreement over Deterministic Broadcast Channels

In this section, we prove Theorem 7.2 which states the secret key generation capacity for the deterministic broadcast channel defined in Section 7.2.2.

7.6.1 Upper Bound for the Key Generation Capacity

Using Theorem 7.5 we know that the secret key generation capacity for independent broadcast channel can be upper bounded as

$$C_s^{\mathsf{det}} \leq \min_{j \in [1:m]} \max_{P_{\boldsymbol{X}_0}} I(\boldsymbol{X}_0; \boldsymbol{X}_j | \boldsymbol{X}_{\mathsf{E}}). \tag{7.51}$$

Then we have the following result, Theorem 7.8.

Theorem 7.8. *The key generation capacity of the deterministic broadcast channel introduced in Section 7.2.2 is upper bounded by* (7.19)*, namely,*

$$C_s^{\mathsf{det}} \leq \sum_{j=1}^{s} [\operatorname{rank} \boldsymbol{F}_j - \operatorname{rank} \boldsymbol{F}_{j-1}] \left(\sum_{i=0}^{j-1} \rho_i \right) \log q, \tag{7.52}$$

where $\rho_i \triangleq \delta_i - 2\delta_i(\delta_0 + \cdots + \delta_{i-1})\mathbb{1}_{\{i>0\}} - \delta_i^2$.

Proof. From (7.51) and because of the symmetry of the problem we have

$$C_s^{\mathsf{det}} \leq \max_{P_{\boldsymbol{X}_\mathsf{A}}} I(\boldsymbol{X}_\mathsf{A}; \boldsymbol{X}_\mathsf{B} | \boldsymbol{X}_\mathsf{E}) = \max_{P_{\boldsymbol{X}_\mathsf{A}}} \left[H(\boldsymbol{X}_\mathsf{A}|\boldsymbol{X}_\mathsf{E}) - H(\boldsymbol{X}_\mathsf{A}|\boldsymbol{X}_\mathsf{B}, \boldsymbol{X}_\mathsf{E}) \right]. \tag{7.53}$$

Then we can write

$$\begin{aligned} H(\boldsymbol{X}_\mathsf{A}|\boldsymbol{X}_\mathsf{E}) &= \sum_{i=0}^{s-1} \delta_i \left[H(\boldsymbol{X}_\mathsf{A}|\widetilde{\boldsymbol{X}}_\mathsf{E}, S_\mathsf{E} = i) \right] \\ &= \sum_{i=0}^{s-1} \delta_i \left[H(\boldsymbol{X}_\mathsf{A}, \widetilde{\boldsymbol{X}}_\mathsf{E}|S_\mathsf{E} = i) - H(\widetilde{\boldsymbol{X}}_\mathsf{E}|S_\mathsf{E} = i) \right] \\ &= \sum_{i=0}^{s-1} \delta_i \left[H(\boldsymbol{X}_\mathsf{A}) - H(\boldsymbol{F}_i \boldsymbol{X}_\mathsf{A}) \right], \end{aligned} \tag{7.54}$$

and similarly

$$H(\boldsymbol{X}_\mathsf{A}|\boldsymbol{X}_\mathsf{B}, \boldsymbol{X}_\mathsf{E}) = \sum_{i=0}^{s-1} \left[2\delta_i(\delta_0 + \cdots + \delta_{i-1})\mathbb{1}_{\{i>0\}} + \delta_i^2 \right] \left[H(\boldsymbol{X}_\mathsf{A}) - H(\boldsymbol{F}_i \boldsymbol{X}_\mathsf{A}) \right]. \tag{7.55}$$

Thus, we have

$$I(\boldsymbol{X}_\mathsf{A}; \boldsymbol{X}_\mathsf{B}|\boldsymbol{X}_\mathsf{E}) = \sum_{i=0}^{s-1} \rho_i \left[H(\boldsymbol{X}_\mathsf{A}) - H(\boldsymbol{F}_i \boldsymbol{X}_\mathsf{A}) \right] \tag{7.56}$$

where $\rho_i \triangleq \delta_i - 2\delta_i(\delta_0 + \cdots + \delta_{i-1})\mathbb{1}_{\{i>0\}} - \delta_i^2$. Now, by knowing that

$$H(\boldsymbol{F}_i \boldsymbol{X}_\mathsf{A}) = H(\boldsymbol{F}_i \boldsymbol{X}_\mathsf{A}, \boldsymbol{F}_{i-1} \boldsymbol{X}_\mathsf{A}) \tag{7.57}$$

and applying the chain rule recursively we have

$$H(\boldsymbol{F}_i \boldsymbol{X}_\mathsf{A}) = \sum_{j=1}^{i} H(\boldsymbol{F}_j \boldsymbol{X}_\mathsf{A} | \boldsymbol{F}_{j-1} \boldsymbol{X}_\mathsf{A}). \tag{7.58}$$

So for $I(\boldsymbol{X}_{\mathsf{A}};\boldsymbol{X}_{\mathsf{B}}|\boldsymbol{X}_{\mathsf{E}})$ we can write

$$\begin{aligned}
I(\boldsymbol{X}_{\mathsf{A}};\boldsymbol{X}_{\mathsf{B}}|\boldsymbol{X}_{\mathsf{E}}) &= \sum_{i=0}^{s-1}\rho_i[H(\boldsymbol{X}_{\mathsf{A}})-H(\boldsymbol{F}_i\boldsymbol{X}_{\mathsf{A}})] \\
&= \sum_{i=0}^{s-1}\rho_i\left[\sum_{j=1}^{s}H(\boldsymbol{F}_j\boldsymbol{X}_{\mathsf{A}}|\boldsymbol{F}_{j-1}\boldsymbol{X}_{\mathsf{A}})-\sum_{j=1}^{i}H(\boldsymbol{F}_j\boldsymbol{X}_{\mathsf{A}}|\boldsymbol{F}_{j-1}\boldsymbol{X}_{\mathsf{A}})\right] \\
&= \sum_{i=0}^{s-1}\rho_i\sum_{j=i+1}^{s}H(\boldsymbol{F}_j\boldsymbol{X}_{\mathsf{A}}|\boldsymbol{F}_{j-1}\boldsymbol{X}_{\mathsf{A}}) \\
&= \sum_{j=1}^{s}H(\boldsymbol{F}_j\boldsymbol{X}_{\mathsf{A}}|\boldsymbol{F}_{j-1}\boldsymbol{X}_{\mathsf{A}})\sum_{i=0}^{j-1}\rho_i. \qquad (7.59)
\end{aligned}$$

Hence we can upper bound C_s^{det} as follows

$$\begin{aligned}
C_s^{\mathsf{det}} &\le \max_{P_{\boldsymbol{X}_{\mathsf{A}}}} I(\boldsymbol{X}_{\mathsf{A}};\boldsymbol{X}_{\mathsf{B}}|\boldsymbol{X}_{\mathsf{E}}) \\
&= \max_{P_{\boldsymbol{X}_{\mathsf{A}}}} \sum_{j=1}^{s}H(\boldsymbol{F}_j\boldsymbol{X}_{\mathsf{A}}|\boldsymbol{F}_{j-1}\boldsymbol{X}_{\mathsf{A}})\sum_{i=0}^{j-1}\rho_i \\
&= \max_{P_{\boldsymbol{X}_{\mathsf{A}}}} \sum_{j=1}^{s}H\left([\boldsymbol{F}_j-\boldsymbol{F}_{j-1}]\boldsymbol{X}_{\mathsf{A}}|\boldsymbol{F}_{j-1}\boldsymbol{X}_{\mathsf{A}}\right)\sum_{i=0}^{j-1}\rho_i \\
&\overset{(a)}{\le} \max_{P_{\boldsymbol{X}_{\mathsf{A}}}} \sum_{j=1}^{s}H\left([\boldsymbol{F}_j-\boldsymbol{F}_{j-1}]\boldsymbol{X}_{\mathsf{A}}\right)\sum_{i=0}^{j-1}\rho_i \\
&\overset{(b)}{=} \sum_{j=1}^{s}\operatorname{rank}\left(\boldsymbol{F}_j-\boldsymbol{F}_{j-1}\right)\left(\sum_{i=0}^{j-1}\rho_i\right)\log q \\
&\overset{(c)}{=} \sum_{j=1}^{s}\left[\operatorname{rank}\boldsymbol{F}_j-\operatorname{rank}\boldsymbol{F}_{j-1}\right]\left(\sum_{i=0}^{j-1}\rho_i\right)\log q, \qquad (7.60)
\end{aligned}$$

where (a) is true because conditioning reduces the entropy, (b) is true because uniform distribution on $\boldsymbol{X}_{\mathsf{A}}$ achieves the maximum values for all the entropies in the summation, and finally (c) is true because of the assumption we have made in (7.12). Also, note that

$$\sum_{i=0}^{j-1}\rho_i = \theta_j(1-\theta_j) \ge 0, \qquad (7.61)$$

where $\theta_j = \sum_{i=0}^{j-1}\delta_i$. □

7.6.2 Lower Bound for the Key Generation Capacity

Because of (7.11) we can find subspaces $\Pi_1, \ldots, \Pi_s$, such that $\Pi_i \cap \Pi_j = \mathbf{0}$ if $i \neq j$ and they also satisfy

$$\begin{aligned} \Pi_1 \oplus \ker \boldsymbol{F}_1 &= \mathbb{F}_q^L, \\ \Pi_2 \oplus \Pi_1 \oplus \ker \boldsymbol{F}_2 &= \mathbb{F}_q^L, \\ &\vdots \\ \Pi_s \oplus \cdots \oplus \Pi_1 \oplus \ker \boldsymbol{F}_s &= \mathbb{F}_q^L. \end{aligned} \tag{7.62}$$

Then for $i \in [1:s]$ we have $\dim \Pi_i = \operatorname{rank} \boldsymbol{F}_i - \operatorname{rank} \boldsymbol{F}_{i-1}$.

In our proposed achievability scheme, Alice uses superposition coding where she creates the vector

$$\boldsymbol{X}_{\mathsf{A}}[t] = \boldsymbol{X}_{\mathsf{A}1}[t] + \cdots + \boldsymbol{X}_{\mathsf{A}s}[t], \tag{7.63}$$

such that $\boldsymbol{X}_{\mathsf{A}i}[t] \in \Pi_i$. Because of (7.62), $\{\Pi_1, \ldots, \Pi_s\}$ form a basis for $\mathbb{F}_q^L$ so every vector $\boldsymbol{X}_{\mathsf{A}}[t] \in \mathbb{F}_q^L$ can be uniquely decomposed as (7.63). Now each $\boldsymbol{X}_{\mathsf{A}i}[t] \in \Pi_i$ can be considered as a vector that is transmitted by Alice and will be received independently by each trusted terminal and Eve with erasure probability

$$\theta_i \triangleq \sum_{j=0}^{i-1} \delta_i. \tag{7.64}$$

Note that the vector $\boldsymbol{X}_{\mathsf{A}i}[t]$ is correctly received by the rth receiver only if $S_r \geq i$.

So we may view the broadcast channel from Alice to the rest of terminals as s independent *packet erasure channels*; where Π_i is the set of messages transmitted over the ith channel (layer) and the erasure probability of the ith channel is θ_i.

Then we proceed as follows. On each layer k, we run independently the scheme propose in Section 7.5.2 for the secret key sharing problem over an erasure broadcast channel. Then we can state the following result, Theorem 7.9.

Theorem 7.9. *The achievable secret key generation rate of the above scheme for each layer k is given by*

$$\mathfrak{R}_k^{\mathsf{det}} = (1-\theta_k)\theta_k \dim(\Pi_k) \log q. \tag{7.65}$$

So for the total achievable secrecy rate we have

$$\begin{aligned} \mathfrak{R}_s^{\mathsf{det}} &= \sum_{j=1}^{s} (1-\theta_j)\theta_j \dim(\Pi_j) \log q \\ &= \sum_{j=1}^{s} [\operatorname{rank} \boldsymbol{F}_j - \operatorname{rank} \boldsymbol{F}_{j-1}] \left(\sum_{i=0}^{j-1} \rho_i \right) \log q, \end{aligned} \tag{7.66}$$

because $(1-\theta_j)\theta_j = \sum_{i=0}^{j-1} \rho_i$ *and* $\dim(\Pi_j) = \operatorname{rank} \boldsymbol{F}_j - \operatorname{rank} \boldsymbol{F}_{j-1}$.

We observe that this matches the upper bound stated in Theorem 7.8, and therefore yields a characterization of the group key-agreement rate for deterministic channels; the result stated in Theorem 7.2.

Remark 7.4. *Similar to the erasure channel problem, this result can also be easily extended to the asymmetric case where the channels to the legitimate users are not statistically identical. Moreover, the key-generation rate is the same for any $m \geq 1$. However, the critical difference between $m = 1$ and $m > 1$ is that the key-reconciliation necessitated the use of ideas from NC.*

7.7 Group Secret Key Agreement over State-dependent Gaussian Broadcast Channels

In this section, we use the results derived in the previous sections in order to study the secret key generation capacity among multiple terminals having access to a state-dependent Gaussian broadcast channel [7].

7.7.1 Upper Bound for the Key Generation Capacity

In order to upper bound the secrecy capacity for the Gaussian broadcast channel, we cannot apply the result of Theorem 7.5 directly because this result has been derived under the assumption that the transmitted and received symbols are discreet. However, the work in [93] has extended the results of [85] for continuous channels. So by using [93, Theorem 6.2], we can write an upper bound for the secrecy capacity similar to Theorem 7.4 with the addition of a power constraint over the transmitted symbols. Then we can state the following result, as stated in Theorem 7.10.

Theorem 7.10. *The key generation capacity of the Gaussian broadcast channel given in* (7.14) *using public discussions is upper bounded as follows*

$$C_s^{\mathsf{gaus}} \leq \frac{1}{2} L \sum_{i=0}^{s} \sum_{j=0}^{s} \delta_i \delta_j \log \left(1 + \frac{h_i^2 P}{1 + h_j^2 P} \right). \tag{7.67}$$

Proof. Using [93, Theorem 6.2] and by proceeding similar steps to the proof of Theorem 7.5, we can write

$$C_s^{\mathsf{gaus}} \leq I(\boldsymbol{X}_{\mathsf{A}}; \boldsymbol{X}_{\mathsf{B}} | \boldsymbol{X}_{\mathsf{E}}) = I(\boldsymbol{X}_{\mathsf{A}}; \widetilde{\boldsymbol{X}}_{\mathsf{B}}, S_{\mathsf{B}} | \widetilde{\boldsymbol{X}}_{\mathsf{E}}, S_{\mathsf{B}}), \tag{7.68}$$

where S_{E} and S_{B} represent the random variables corresponding to Eve's and Bob's channel states respectively. Hence, we can write

$$\begin{aligned} C_s^{\mathsf{gaus}} &\leq I(\boldsymbol{X}_{\mathsf{A}}; \boldsymbol{X}_{\mathsf{B}} | \boldsymbol{X}_{\mathsf{E}}) \\ &= H(\widetilde{\boldsymbol{X}}_{\mathsf{B}}, S_{\mathsf{B}} | \widetilde{\boldsymbol{X}}_{\mathsf{E}}, S_{\mathsf{E}}) - H(\widetilde{\boldsymbol{X}}_{\mathsf{B}}, S_{\mathsf{B}} | \widetilde{\boldsymbol{X}}_{\mathsf{E}}, S_{\mathsf{E}}, \boldsymbol{X}_{\mathsf{A}}) \end{aligned}$$

7. In this section, with an abuse of notation, we use $H(\cdot)$ to denote for the differential entropy as well.

$$
\begin{aligned}
&\overset{(a)}{=} H(\widetilde{\boldsymbol{X}}_{\mathsf{B}}, S_{\mathsf{B}}|\widetilde{\boldsymbol{X}}_{\mathsf{E}}, S_{\mathsf{E}}) - H(\widetilde{\boldsymbol{X}}_{\mathsf{B}}, S_{\mathsf{B}}|\boldsymbol{X}_{\mathsf{A}}) \\
&= H(\widetilde{\boldsymbol{X}}_{\mathsf{B}}, S_{\mathsf{B}}|\widetilde{\boldsymbol{X}}_{\mathsf{E}}, S_{\mathsf{E}}) - H(S_{\mathsf{B}}|\boldsymbol{X}_{\mathsf{A}}) - H(\widetilde{\boldsymbol{X}}_{\mathsf{B}}|S_{\mathsf{B}}, \boldsymbol{X}_{\mathsf{A}}) \\
&\overset{(b)}{=} H(\widetilde{\boldsymbol{X}}_{\mathsf{B}}, S_{\mathsf{B}}|\widetilde{\boldsymbol{X}}_{\mathsf{E}}, S_{\mathsf{E}}) - H(S_{\mathsf{B}}) - H(\boldsymbol{Z}_{\mathsf{B}}) \\
&= H(\widetilde{\boldsymbol{X}}_{\mathsf{B}}, \widetilde{\boldsymbol{X}}_{\mathsf{E}}|S_{\mathsf{E}}, S_{\mathsf{B}}) + H(S_{\mathsf{E}}, S_{\mathsf{B}}) \\
&\quad - H(\widetilde{\boldsymbol{X}}_{\mathsf{E}}, S_{\mathsf{E}}) - H(S_{\mathsf{B}}) - H(\boldsymbol{Z}_{\mathsf{B}}) \\
&= H(\widetilde{\boldsymbol{X}}_{\mathsf{B}}, \widetilde{\boldsymbol{X}}_{\mathsf{E}}|S_{\mathsf{E}}, S_{\mathsf{B}}) - H(\widetilde{\boldsymbol{X}}_{\mathsf{E}}|S_{\mathsf{E}}) - H(\boldsymbol{Z}_{\mathsf{B}}) \\
&= \sum_{i=0}^{s}\sum_{j=0}^{s} \delta_i\delta_j H(\widetilde{\boldsymbol{X}}_{\mathsf{B}}, \widetilde{\boldsymbol{X}}_{\mathsf{E}}|S_{\mathsf{E}} = j, S_{\mathsf{B}} = i) \\
&\quad - \sum_{k=0}^{s} \delta_k H(\widetilde{\boldsymbol{X}}_{\mathsf{E}}|S_{\mathsf{E}} = k) - H(\boldsymbol{Z}_{\mathsf{B}}) \\
&= \sum_{i=0}^{s}\sum_{j=0}^{s} \delta_i\delta_j H(h_i\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{B}}, h_j\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{E}}) \\
&\quad - \sum_{k=0}^{s} \delta_k H(h_k\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{E}}) - H(\boldsymbol{Z}_{\mathsf{B}}) \\
&= \sum_{i=0}^{s}\sum_{j=0}^{s} \delta_i\delta_j H(h_i\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{B}}|h_j\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{E}}) - H(\boldsymbol{Z}_{\mathsf{B}}) \\
&\overset{(c)}{\leq} \sum_{i=0}^{s}\sum_{j=0}^{s} \frac{\delta_i\delta_j}{2} \log\left[(2\pi e)^L(\mathrm{cov}(h_i\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{B}}|h_j\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{E}}))\right] - H(\boldsymbol{Z}_{\mathsf{B}}),
\end{aligned}
\tag{7.69}
$$

where (a) is true since we have the Markov chain $\boldsymbol{X}_{\mathsf{B}} \leftrightarrow \boldsymbol{X}_{\mathsf{A}} \leftrightarrow \boldsymbol{X}_{\mathsf{E}}$, (b) follows from the fact that the state variables are independent of $\boldsymbol{X}_{\mathsf{A}}$ and given $\boldsymbol{X}_{\mathsf{A}}$ and S_{B} the only uncertainty left in $\widetilde{\boldsymbol{X}}_{\mathsf{B}}$ is that of noise $\boldsymbol{Z}_{\mathsf{B}}$ and finally (c) follows from the fact that for a fixed variance, Gaussian distribution maximizes the entropy.

The inequality (c) in (7.69) is achieved when $(h_i\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{B}}|h_j\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{E}})$ has a Gaussian distribution. A sufficient condition for this to be satisfied is when $\boldsymbol{X}_{\mathsf{A}}$, $\boldsymbol{Z}_{\mathsf{B}}$, and $\boldsymbol{Z}_{\mathsf{E}}$ are Gaussian and independent. This observation makes the calculation of

$$
\frac{1}{2}\log\left[(2\pi e)^L\,\mathrm{cov}(h_i\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{B}}|h_j\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{E}})\right] \tag{7.70}
$$

much easier as it is equivalent to the evaluation of $H(h_i\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{B}}, h_j\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{E}}) - H(h_j\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{E}})$ when $\boldsymbol{X}_{\mathsf{A}}$, $\boldsymbol{Z}_{\mathsf{B}}$, and $\boldsymbol{Z}_{\mathsf{E}}$ are Gaussian and independent as shown below,

$$
\begin{aligned}
&\frac{1}{2}\log\left[(2\pi e)^L\,\mathrm{cov}(h_i\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{B}}|h_j\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{E}})\right] = \\
&\qquad = H\big(h_i\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{B}}, h_j\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{E}}\big) - H\big(h_j\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{E}}\big)
\end{aligned}
$$

$$= \sum_{k=1}^{L} H\big(h_i X_{\mathsf{A},k} + Z_{\mathsf{B},k}, h_j X_{\mathsf{A},k} + Z_{\mathsf{E},k}\big) - H\big(h_j X_{\mathsf{A},k} + Z_{\mathsf{E},k}\big)$$
$$= \frac{L}{2}\left[\log\big((2\pi e)^2(1+h_i^2P+h_j^2P)\big) - \log\big(2\pi e(1+h_j^2P)\big)\right], \quad (7.71)$$

where $\mathbb{E}\left[X_{\mathsf{A},k}^2\right] = P$ and $\mathbb{E}\left[Z_{\mathsf{B},k}^2\right] = \mathbb{E}\left[Z_{\mathsf{E},k}^2\right] = 1$ for all $k \in [1:L]$.

Hence, the upper bound on the secrecy capacity becomes as follows

$$\begin{aligned}
C_s^{\mathsf{gaus}} &\le I(\boldsymbol{X}_{\mathsf{A}}; \boldsymbol{X}_{\mathsf{B}}|\boldsymbol{X}_{\mathsf{E}}) \\
&\le \sum_{i=0}^{s}\sum_{j=0}^{s} \frac{\delta_i\delta_j}{2} \log\big((2\pi e)^L \operatorname{var}(h_i\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{B}}|h_j\boldsymbol{X}_{\mathsf{A}} + \boldsymbol{Z}_{\mathsf{E}})\big) - H(\boldsymbol{Z}_{\mathsf{B}}) \\
&= \sum_{i=0}^{s}\sum_{j=0}^{s} \frac{\delta_i\delta_j L}{2} \log\big((2\pi e)^2(1+h_i^2P+h_j^2P)\big) \\
&\quad - \sum_{i=0}^{s}\sum_{j=0}^{s} \frac{\delta_i\delta_j L}{2} \log\big(2\pi e(1+h_j^2P)\big) - \frac{L}{2}\log(2\pi e) \\
&= \sum_{i=0}^{s}\sum_{j=0}^{s} \frac{\delta_i\delta_j L}{2} \left[\log(1+h_i^2P+h_j^2P) - \log(1+h_j^2P)\right] \\
&= \frac{1}{2}L\sum_{i=0}^{s}\sum_{j=0}^{s} \delta_i\delta_j \log\left(1 + \frac{h_i^2P}{1+h_j^2P}\right), \qquad (7.72)
\end{aligned}$$

where we are done. □

7.7.2 Lower Bound for the Key Generation Capacity

Before giving the achievability scheme, let us define a nested message set, degraded channel wiretap scenario as follows.

Definition 7.3. *Assume a wiretap channel scenario where there is a transmitter called Alice that broadcasts X_{A} and there are $s+1$ receivers Y_i where the ith receiver receives Y_i according to the broadcast channel $(\mathcal{X}_{\mathsf{A}}, p(y_0,\ldots,y_s|x), \mathcal{Y}_0 \times \cdots \times \mathcal{Y}_s))$ such that*

$$p(y_0,\ldots,y_s|x_{\mathsf{A}}) = p(y_s|x_{\mathsf{A}}) \cdot p(y_{s-1}|y_s)\cdots p(y_0|y_1). \quad (7.73)$$

Suppose that Alice has s messages $W_1,\ldots,W_s$ where $W_i \in \{1,\ldots,2^{LR_i}\}$ and $W_i \sim \mathsf{Uni}(1:2^{LR_i})$. The goal is that she wants to broadcast these messages such that $\forall i$:
(i) *each message W_i should be decodable by the receivers $Y_i,\ldots,Y_s$ with a negligible error probability, and*
(ii) *all the receivers $Y_0,\ldots,Y_{i-1}$ should be ignorant about the message W_i, namely for the leakage rate we have*

$$R_{l,i}^{(L)} \triangleq \frac{1}{L} I(W_{i+1},\ldots,W_s; Y_i^{1:L}) \le \epsilon_L, \forall i \in [0:s]. \quad (7.74)$$

Then we can state the following result.

Theorem 7.11. *Using a properly designed layered wiretap code similar to [90] we can achieve the following set of rates for the nested message set, degraded Gaussian wiretap channel*

$$R_i = \frac{1}{2}\left[\log\left(1+\frac{h_i^2 P_i}{1+h_i^2 I_i}\right) - \log\left(1+\frac{h_{i-1}^2 P_i}{1+h_{i-1}^2 I_i}\right)\right], \tag{7.75}$$

$\forall i \in [1:s]$, *where* $I_i \triangleq \sum_{j=i+1}^{s} P_j$.

By using a layered coding scheme for the nested message set, degraded channel wiretap channel defined in Definition 7.3, we can convert the Gaussian channel given in (7.14) to a set of s independent erasure channels where the erasure of the messages for each channel (layer) depends on the receiver channel state. In fact using the layered coding scheme for the wiretap channel, we mimic the orthogonality behavior that we have for the deterministic channel as described by (7.11) and (7.62).

More precisely, we assume that Alice broadcasts the L-length vector

$$\boldsymbol{X}_{\mathsf{A}}[t] = \sum_{i=1}^{s} \boldsymbol{X}_{\mathsf{A}i}[t], \tag{7.76}$$

where she maps W_i (the messages corresponding to the ith layer) to $\boldsymbol{X}_{\mathsf{A}i}[t]$ according to the codebook described in the following. We construct s codebooks $\hat{\mathcal{C}}_i(2^{L\hat{R}_i}, L)$ each contains $2^{L\hat{R}_i}$ codewords $X_{\mathsf{A}i}^L$ by choosing L symbols independently from the Gaussian distribution $\mathcal{N}(0, P_i)$ where

$$\hat{R}_i = \frac{1}{2}\log\left(1+\frac{h_i^2 P_i}{1+h_i^2 I_i}\right). \tag{7.77}$$

Each codebook $\hat{\mathcal{C}}_i$, $i \in [1:s]$, is divided into 2^{LR_i} bins where R_i is given by (7.75). At each layer i, the message W_i is coded so as to be secure from all receivers in states $j < i$. This is done by a standard wiretap code (see also [90]), where the message W_i is the bin-index and the transmit sequence $\boldsymbol{X}_{\mathsf{A}i}$ is a (random) sequence from the the bin. So, the ith layer can transmits 2^{LR_i} messages securely from the "weaker" receivers. Following a similar argument as stated in [90], we can show that the receiver r which observes the channel state $S_r = i$ can decode messages up to layer i and is ignorant about messages of layers above i. So, equivalently, we can say that the message W_i experiences erasure probability

$$\theta_i = \sum_{j=0}^{i-1} \delta_j, \tag{7.78}$$

when it passes through the channel (7.14).

Now for each layer i, we run the interactive secret key sharing scheme introduced in Section 7.5.2 where Alice broadcasts a sequence of random messages

W_i^n. Then, by discussing over the public channel, the trusted terminals reconcile their secret messages to build a common key. The key generation rate for each layer is $\Delta_i LR_i$, so for a fixed power allocation we achieve the following secrecy rate

$$\mathfrak{R}_s^{\mathsf{gaus}} \leq \sum_{i=1}^{s} \Delta_i LR_i, \tag{7.79}$$

where R_i is defined in (7.75) and $\Delta_i \triangleq (1-\theta_i)\theta_i$.

The maximum secrecy rate is obtained by optimizing the above rate over the power allocations $\{P_i\}_{i=1}^s$. Thus we can write

$$\mathfrak{R}_s^{\mathsf{gaus}} = \left\{ \begin{array}{ll} \max & \sum_{i=1}^{s} \Delta_i LR_i \\ \text{subject to} & \sum_{i=1}^{s} P_i \leq P \\ & P_i \geq 0, \quad \forall i \in [1:s]. \end{array} \right. \tag{7.80}$$

Because R_1 is an increasing function of P_1 when other P_i are kept fixed and R_i does not depend on P_1 for $i > 1$ we can write the power constant inequality as an equality. We also apply a change of variables to Program (7.80) from $\{P_i\}$ to $\{I_k\}$. So we can rewrite Program (7.80) as follows

$$\mathfrak{R}_s^{\mathsf{gaus}} = \left\{ \begin{array}{ll} \min & -\sum_{i=1}^{s} \Delta_i LR_i \\ \text{subject to} & -[I_{k-1} - I_k] \leq 0, \ \forall k \in [1:s], \end{array} \right. \tag{7.81}$$

where $I_0 = P$, $I_s = 0$, and we have

$$R_i = \frac{1}{2} \log \left(\frac{1 + h_i^2 I_{i-1}}{1 + h_i^2 I_i} \cdot \frac{1 + h_{i-1}^2 I_i}{1 + h_{i-1}^2 I_{i-1}} \right). \tag{7.82}$$

Because the constraints of the Program (7.81) are affine we can apply the KKT conditions (see [94, Proposition 5.4.1]) to obtain a necessary condition for optimum power allocation. By defining the Lagrangian $\mathcal{L}$ as

$$\mathcal{L}(P_1, \ldots, P_s, \lambda_1, \ldots, \lambda_s) = -\sum_{i=1}^{s} \Delta_i LR_i + \sum_{i=1}^{s} \lambda_i [I_i - I_{i-1}], \tag{7.83}$$

we may write a set of necessary conditions to maximize R_s as follows

$$\left\{ \begin{array}{ll} \frac{\partial \mathcal{L}}{\partial I_k} = 0, & \forall k \in [1:s-1], \\ \lambda_k [I_k - I_{k-1}] = 0, & \forall k \in [1:s], \\ I_k \leq I_{k-1}, & \forall k \in [1:s], \\ \lambda_k \geq 0, & \forall k \in [1:s]. \end{array} \right. \tag{7.84}$$

In general solving analytically the set of conditions stated in (7.84) is a tedious task (for more discussions on this refer to Section 7.B). However, it is possible to reformulate the optimization problem (7.81) as a *Generalized Linear Fractional Program* (see Appendix 7.C for some basic definitions), and then by applying the results of [95] and [96], it is possible to find numerically the

optimal power allocation for each layer (see Appendices 7.C and 7.D for more discussion).

Moreover, one of the important special case of this problem is when there is a large dynamic range between the channel states, and we focus on this case applied to the high SNR regime. This enables us to demonstrate an optimal power allocation for this regime in Section 7.7.3.

7.7.3 High SNR Regime

By large dynamic range in the states, we mean that $h_i \gg h_{i-1}$, $\forall i \in [1:s]$, where this comparison is done with respect to SNR. In particular, we denote

$$h_i^2 = \mathrm{SNR}^{-\alpha_i}, \qquad i \in [0:s], \tag{7.85}$$

where $h_i^2 > h_{i-1}^2$ implying that $\alpha_i < \alpha_{i-1}$. Suppose we apply a power allocation which is given as follows, $P_i = \mathrm{SNR}^{\beta_i}$, $\forall i \in [1:s]$, with the assumption that $\beta_i < \beta_{i-1}$. Since $\beta_i < \beta_{i-1}$, in high SNR regime I_i is dominated by $\mathrm{SNR}^{\beta_{i+1}}$. Using this approximation, we can rewrite the expression for R_i from (7.75) as follows

$$\begin{aligned} R_i &\doteq \frac{1}{2}\log\left(1+\frac{\mathrm{SNR}^{\beta_i-\alpha_i}}{1+\mathrm{SNR}^{\beta_{i+1}-\alpha_i}}\right) - \frac{1}{2}\log\left(1+\frac{\mathrm{SNR}^{\beta_i-\alpha_{i-1}}}{1+\mathrm{SNR}^{\beta_{i+1}-\alpha_{i-1}}}\right) \\ &\doteq \frac{1}{2}\left[\left((\beta_i-\alpha_i)-(\beta_{i+1}-\alpha_i)^+\right)^+\right]\log\mathrm{SNR} \\ &\quad - \frac{1}{2}\left[\left((\beta_i-\alpha_{i-1})-(\beta_{i+1}-\alpha_{i-1})^+\right)^+\right]\log\mathrm{SNR}, \end{aligned} \tag{7.86}$$

where we use the notation "$\dot{=}$" and "$\dot{\leq}$" for exponential equality and inequality with respect to SNR. Using the power allocation $\beta_1 = 1-\epsilon$ and $\beta_i = \alpha_{i-1} - \epsilon$, $\forall i \in [2:s]$, with $\epsilon > 0$ and $\epsilon \ll 1$ we can write

$$R_i \doteq (\alpha_{i-1} - \epsilon - \alpha_i)\log\mathrm{SNR} \doteq (\alpha_{i-1}-\alpha_i)\log\mathrm{SNR}. \tag{7.87}$$

Intuitively $\beta_i = \alpha_{i-1}$ can be interpreted as a power allocation *matching* the channel gains. So, for the total achievable secrecy rate we have

$$\mathfrak{R}_s^{\mathsf{gaus}} \doteq \frac{1}{2}L\sum_{i=1}^{s}\Delta_i(\alpha_{i-1}-\alpha_i)\log\mathrm{SNR} = \frac{1}{2}L\sum_{i=1}^{s}\Delta_i\log\frac{h_i^2}{h_{i-1}^2}. \tag{7.88}$$

Now, we state the upper bound to C_s^{gaus} in the high SNR regime in Theorem 7.12.

Theorem 7.12. *Assuming high-SNR regime and large dynamic range over channel states we can upper bound C_s^{gaus} as follows*

$$C_s^{\mathsf{gaus}} \dot{\leq} \frac{1}{2}L\sum_{i=1}^{s}\Delta_i(\alpha_{i-1}-\alpha_i)\log\mathrm{SNR}. \tag{7.89}$$

This upper bound is matched with the achievable rate derived in (7.88)*, so the above equation characterizes* C_s^{gaus} *in this regime.*

Proof. Let us compute the upper bound to C_s^{gaus} in the high-SNR regime as follows. From Theorem 7.10 and assuming that $h_i^2 = \mathrm{SNR}^{-\alpha_i}$ we can write

$$\begin{aligned}
C_s^{\mathsf{gaus}} &\leq \frac{1}{2} L \sum_{i=0}^{s} \sum_{j=0}^{s} \delta_i \delta_j \left[\log(1 + h_i^2 P + h_j^2 P) - \log(1 + h_j^2 P) \right] \\
&\doteq \frac{1}{2} L \sum_{i>j} \delta_i \delta_j (\alpha_j - \alpha_i) \log \mathrm{SNR} \\
&= \frac{1}{2} L \sum_{i=1}^{s} \sum_{j=0}^{i-1} (\alpha_j - \alpha_i) \delta_i \delta_j \log \mathrm{SNR} \\
&= \frac{1}{2} L \sum_{i=1}^{s} \sum_{j=0}^{i-1} \left(\sum_{k=j+1}^{i} (\alpha_{k-1} - \alpha_k) \right) \delta_i \delta_j \log \mathrm{SNR} \\
&= \frac{1}{2} L \sum_{i=1}^{s} \sum_{j=0}^{i-1} \sum_{k=j+1}^{i} (\alpha_{k-1} - \alpha_k) \delta_i \delta_j \log \mathrm{SNR} \\
&\overset{\text{(a)}}{=} \frac{1}{2} L \sum_{k=1}^{s} \sum_{i=k}^{s} \sum_{j=0}^{k-1} (\alpha_{k-1} - \alpha_k) \delta_i \delta_j \log \mathrm{SNR} \\
&= \frac{1}{2} L \sum_{k=1}^{s} \Delta_k (\alpha_{k-1} - \alpha_k) \log \mathrm{SNR}, \qquad (7.90)
\end{aligned}$$

where (a) simply follows by exchanging the order of the summations and we are done. □

7.8 Concluding Remarks

In this chapter, we have considered the problem of secret key sharing among multiple trusted terminals having access to a wireless broadcast channel in the presence of a passive eavesdropper Eve. We assume that Eve also observes a noisy signal from the broadcast channel. Moreover, we assume that the legitimate terminals can publicly discuss over a rate-unlimited public channel which is also overheard by Eve.

For the aforementioned scenario, we have characterized the secret key sharing capacity and proposed efficient (polynomial-time algorithms) schemes to achieve it for the case of erasure as well as state-dependent deterministic broadcast channels.

For the case of state-dependent Gaussian broadcast channel, we have used a nested-set degraded channel wiretap code to mimic the orthogonality of different layers we had for the deterministic broadcast channel. Then, we converted

the problem to multiple parallel erasure channels and applied the scheme developed for the erasure channel on every layer (channel) independently. By doing so, we have proposed an achievability scheme for the state-dependent Gaussian broadcast channel. Although the achievable secrecy rate is not matched the upper bound we have, for the high dynamic range and high SNR regime we have shown that they match in the sense of degrees of freedom.

7.A Some Lemmas

Lemma 7.1. *Consider a set of n packets $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, $\boldsymbol{x}_i \in \mathbb{F}_q^L$, where $\boldsymbol{x}_i \sim \mathsf{Uni}\left(\mathbb{F}_q^L\right)$ and all packets $\boldsymbol{x}_i$ are independent from each other. Assume that Eve has overheard n_{E} of these packets. Call the packets Eve has $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{n_{\mathsf{E}}}$. Then it is possible to create $n' = n - n_{\mathsf{E}}$ linear combinations of the $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ packets over the finite field $\mathbb{F}_q$, say $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n'}$, in polynomial time, so that these are secure from Eve, i.e.,*

$$I(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n'}; \boldsymbol{w}_1, \ldots, \boldsymbol{w}_{n_{\mathsf{E}}}) = 0. \tag{7.91}$$

The same result holds with high probability (of order $1 - O(q^{-1})$) if the linear combinations are selected uniformly at random over $\mathbb{F}_q$.

Proof. Let us construct a matrix $\boldsymbol{X}$ that has as rows the packets $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. Similarly, construct matrices $\boldsymbol{Y}$ and $\boldsymbol{W}$ that have as rows the packets $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n'}$ and $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{n_{\mathsf{E}}}$.

Note that because the packets $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{n_{\mathsf{E}}}$ are by definition a subset of the packets $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, we can write $\boldsymbol{W} = \boldsymbol{A}_{\mathsf{E}}\boldsymbol{X}$, with $\boldsymbol{A}_{\mathsf{E}} \in \mathbb{F}_q^{n_{\mathsf{E}} \times n}$ that has zeros and ones as elements. We will also construct the $\boldsymbol{y}$-packets as linear combinations of the $\boldsymbol{x}$-packets over a field $\mathbb{F}_q$. We will then have that $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$, where $\boldsymbol{A} \in \mathbb{F}_q^{(n-n_{\mathsf{E}}) \times n}$ is the matrix we are interested in designing. Thus we can write

$$\begin{bmatrix} \boldsymbol{Y} \\ \boldsymbol{W} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A}_{\mathsf{E}} \end{bmatrix} \boldsymbol{X}. \tag{7.92}$$

We now proceed by expanding $H(\boldsymbol{Y}|\boldsymbol{W})$. We have

$$\begin{aligned} H(\boldsymbol{Y}|\boldsymbol{W}) &= H(\boldsymbol{Y}, \boldsymbol{W}) - H(\boldsymbol{W}) \\ &= \big[\operatorname{rank}(\boldsymbol{B}) - \operatorname{rank}(\boldsymbol{A}_{\mathsf{E}})\big] L \log q \\ &= \big[\operatorname{rank}(\boldsymbol{B}) - n_{\mathsf{E}}\big] L \log q, \end{aligned} \tag{7.93}$$

where $\boldsymbol{B} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A}_{\mathsf{E}} \end{bmatrix}$ and L is the length of each packet $\boldsymbol{x}_i$. Now the only way that we have $H(\boldsymbol{Y}|\boldsymbol{W}) = H(\boldsymbol{Y})$ is that $\boldsymbol{B}$ becomes a full rank matrix.

Using coding theory we will construct such a matrix $\boldsymbol{B}$, *without* knowing $\boldsymbol{A}_{\mathsf{E}}$. All we know is that in each row of $\boldsymbol{A}_{\mathsf{E}}$ there is only one 1 and the remaining elements are zero; so all of the vectors in the row span of $\boldsymbol{A}_{\mathsf{E}}$ have Hamming weight less than or equal to n_{E}. Now, if we choose $\boldsymbol{A}$ to be a generator matrix of an maximum distance separable (MDS) linear code with parameters $[n, n - n_{\mathsf{E}}, n_{\mathsf{E}} + 1]_q$ then each codeword has Hamming weight larger than or equal to $n_{\mathsf{E}} + 1$ [48]. So the row span of $\boldsymbol{A}$ and $\boldsymbol{A}_{\mathsf{E}}$ are disjoint (except for the zero vector) and the matrix $\boldsymbol{B}$ becomes full-rank for all of matrices $\boldsymbol{A}_{\mathsf{E}}$ that have the aforementioned structure. For example, we may select to use a generator matrix of a Reed-Solomon code [48], which is an MDS code, over a field of size $q = n + 1$.

To prove the second assertion of the lemma, we note that creating vectors $\boldsymbol{y}_i$ uniformly at random is equivalent to selecting the elements of matrix $\boldsymbol{A}$ independently uniformly at random from the field $\mathbb{F}_q$. In this case we can write

$$\begin{aligned}\mathbb{P}\left[\boldsymbol{B} \text{ is full-rank}\right] &= \frac{(q^n - q^{n_{\mathsf{E}}}) \cdots (q^n - q^{n-1})}{q^{n(n-n_{\mathsf{E}})}} \\ &= \left(1 - q^{-(n-n_{\mathsf{E}})}\right) \cdots (1 - q^{-1}) \\ &= 1 - O(q^{-1}),\end{aligned} \tag{7.94}$$

which goes to 1 as q increases. □

Lemma 7.2. *Consider packets $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_h$ and assume that each one of $m-1$ receivers has observed a different subset of these packets of size l. We can find $h - l$ linear combinations of the $\boldsymbol{y}$-packets, say $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{h-l}$ such that, each receiver can use its observations and the $\boldsymbol{z}$-packets to decode all the $\boldsymbol{y}$-packets.*

Proof. This is a standard problem formulation in the NC literature, and any of the standard polynomial-time approaches for network code design can be used [8]. □

Lemma 7.3. *Consider a set of h packets $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_h$ where $\boldsymbol{y}_i \sim \mathsf{Uni}\left(\mathbb{F}_q^L\right)$ and assume that an eavesdropper Eve has overheard linear combinations of $h - l$ of these packets. Call the packets Eve has $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{h-l}$. Then it is possible to create l linear combinations of the $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_h$ packets, say $\boldsymbol{k}_1, \ldots, \boldsymbol{k}_l$, in polynomial time, so that these are secure from Eve, i.e.,*

$$I(\boldsymbol{k}_1, \ldots, \boldsymbol{k}_l; \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{h-l}) = 0. \tag{7.95}$$

The same result holds with high probability (probability of order $1 - O(q^{-1})$) if the l packets $\boldsymbol{k}_i$ are created uniformly at random over $\mathbb{F}_q$.

Proof. Similar to the proof of Lemma 7.1, let $\boldsymbol{Y}$, $\boldsymbol{Z}$ and $\boldsymbol{K}$ be matrices that have as rows the packets $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_h$, $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{h-l}$ and $\boldsymbol{k}_1, \ldots, \boldsymbol{k}_l$. We can then write

$$\begin{bmatrix} \boldsymbol{K} \\ \boldsymbol{Z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_K \\ \boldsymbol{A}_Z \end{bmatrix} \boldsymbol{Y}, \tag{7.96}$$

where $\boldsymbol{A}_Z$ is a given known matrix, since we know the transmitted linear combinations, and we seek a matrix $\boldsymbol{A}_K$ such that, the matrix $\begin{bmatrix} \boldsymbol{A}_K \\ \boldsymbol{A}_Z \end{bmatrix}$ is full rank. Equivalently, we seek vectors $\boldsymbol{k}_1, \ldots, \boldsymbol{k}_l$ that together with $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{h-l}$ form a basis; we can do this using any of standard methods, such as Gram-Schmidt orthogonalization. □

Lemma 7.4. *The value of the parameter l in Theorem 7.7 converges exponentially fast in n to its expected value.*

Proof. Let us consider the *random* variables h, h_i, and l defined in Section 7.5.2. For convenience, we will work with the normalized random variables $\overline{h} \triangleq h/n$, $\overline{h}_i \triangleq h_i/n$, and $\overline{l} \triangleq l/n$. Let us also define the random variable $\eta_j^{(i)}$ as follows

$$\eta_j^{(i)} = \begin{cases} 1 & \text{if the } j\text{th } x\text{-packet is received} \\ & \text{by } \mathsf{T}_i \text{ but not by Eve,} \\ 0 & \text{otherwise.} \end{cases} \tag{7.97}$$

Then we can write $\overline{h}_i = \frac{1}{n}\sum_{j=1}^{n} \eta_j^{(i)}$ and we have $\mu = \mu_i \triangleq \mathbb{E}\left[\overline{h}_i\right] = (1-\delta)\delta_E$. As defined before, we have also $\overline{l} = \min_i \overline{h}_i$.

To bound $\overline{l}$, observe that for some small γ, $0 < \gamma \leq \mu$, we can write

$$\begin{aligned} \mathbb{P}\left[\overline{l} \geq \mu + \gamma\right] &= \mathbb{P}\left[\overline{h}_i \geq \mu + \gamma : \forall i\right] \\ &= \mathbb{P}\left[\overline{h}_1 \geq \mu + \gamma\right]^m \\ &\leq \exp\left(-\frac{m\gamma^2}{3\mu} n\right), \end{aligned} \tag{7.98}$$

where in the last inequality we use Chernoff bound [97, Chapter 4]. On the other hand we can also write for $0 < \gamma \leq \mu$

$$\begin{aligned} \mathbb{P}\left[\overline{l} \leq \mu - \gamma\right] &\leq m\mathbb{P}\left[\overline{h}_1 \leq \mu - \gamma\right] \\ &\leq m \exp\left(-\frac{\gamma^2}{2\mu} n\right), \end{aligned} \tag{7.99}$$

so we are done. □

7.B Discussion on the Power Allocation Optimization Problem

This section is an attempt to solve the set of necessary conditions (7.84). We do not provide an analytical solution for the general case, however we can derive the optimum power allocation for some special cases.

Recall that the Lagrangian $\mathcal{L}$ of the optimization problem (7.81) is as follows

$$\mathcal{L}(P_1, \ldots, P_s, \lambda_1, \ldots, \lambda_s) = -\sum_{i=1}^{s} \Delta_i R_i + \sum_{i=1}^{s} \lambda_i [I_i - I_{i-1}]. \tag{7.100}$$

For more convenience, let us define $\mathfrak{h}_i \triangleq h_i^2$. Then by taking the derivative of $\mathcal{L}$ with respect to I_k we have

$$\begin{aligned} (\ln 2)\frac{\partial \mathcal{L}}{\partial I_k} &= -\frac{(\mathfrak{h}_{k+1} - \mathfrak{h}_k)\Delta_{k+1}}{(1+\mathfrak{h}_k I_k)(1+\mathfrak{h}_{k+1} I_k)} + \frac{(\mathfrak{h}_k - \mathfrak{h}_{k-1})\Delta_k}{(1+\mathfrak{h}_{k-1} I_k)(1+\mathfrak{h}_k I_k)} \\ &\quad + (\ln 2)(\lambda_k - \lambda_{k+1}) \\ &= \frac{\Delta_k(\mathfrak{h}_k - \mathfrak{h}_{k-1})(1+\mathfrak{h}_{k+1} I_k) - \Delta_{k+1}(\mathfrak{h}_{k+1} - \mathfrak{h}_k)(1+\mathfrak{h}_{k-1} I_k)}{(1+\mathfrak{h}_{k-1} I_k)(1+\mathfrak{h}_k I_k)(1+\mathfrak{h}_{k+1} I_k)} \\ &\quad + (\ln 2)(\lambda_k - \lambda_{k+1}). \end{aligned} \tag{7.101}$$

We can further expand the derivative of $\mathcal{L}$ as follows

$$\begin{aligned}(\ln 2)\frac{\partial \mathcal{L}}{\partial I_k} &= \frac{[\mathfrak{h}_{k+1}(\mathfrak{h}_k-\mathfrak{h}_{k-1})\Delta_k - \mathfrak{h}_{k-1}(\mathfrak{h}_{k+1}-\mathfrak{h}_k)\Delta_{k+1}]}{(1+\mathfrak{h}_{k-1}I_k)(1+\mathfrak{h}_kI_k)(1+\mathfrak{h}_{k+1}I_k)}I_k \\ &\quad - \frac{[(\mathfrak{h}_{k+1}-\mathfrak{h}_k)\Delta_{k+1} - (\mathfrak{h}_k-\mathfrak{h}_{k-1})\Delta_k]}{(1+\mathfrak{h}_{k-1}I_k)(1+\mathfrak{h}_kI_k)(1+\mathfrak{h}_{k+1}I_k)} + (\ln 2)(\lambda_k-\lambda_{k+1}) \\ &= \frac{(\mathfrak{h}_{k+1}\beta_k - \mathfrak{h}_{k-1}\alpha_k)I_k - (\alpha_k-\beta_k)}{(1+\mathfrak{h}_{k-1}I_k)(1+\mathfrak{h}_kI_k)(1+\mathfrak{h}_{k+1}I_k)} + (\ln 2)(\lambda_k-\lambda_{k+1}) \\ &= F_k(I_k) + (\ln 2)(\lambda_k-\lambda_{k+1}), \end{aligned} \tag{7.102}$$

where $\alpha_k \triangleq (\mathfrak{h}_{k+1}-\mathfrak{h}_k)\Delta_{k+1}$ and $\beta_k \triangleq (\mathfrak{h}_k-\mathfrak{h}_{k-1})\Delta_k$. Let us define

$$\tilde{I}_k \triangleq \frac{\alpha_k-\beta_k}{\mathfrak{h}_{k+1}\beta_k - \mathfrak{h}_{k-1}\alpha_k}, \qquad \forall k \in [1:s-1], \tag{7.103}$$

and by convention we set $\tilde{I}_0 = P$ and $\tilde{I}_s = 0$.

Based on the values of α_k and β_k we have different situations as follows:

1. If $\alpha_k < \beta_k$ we can conclude that $\mathfrak{h}_{k+1}\beta_k > \mathfrak{h}_{k-1}\alpha_k$ so we have $\tilde{I}_k < 0$ and $F_k(x) > 0$ for $x \geq 0$. Because we should have the condition $F_k(I_k^*) + (\ln 2)(\lambda_k^* - \lambda_{k+1}^*) = 0$ satisfied for the optimum solution we conclude that $\lambda_{k+1}^* > 0$ which using the complementary slackness condition results in $I_k^* = I_{k+1}^*$.
2. If $\mathfrak{h}_{k+1}\beta_k < \mathfrak{h}_{k-1}\alpha_k$ we can conclude that $\alpha_k > \beta_k$ so we have $\tilde{I}_k < 0$ and $F_k(x) < 0$ for $x \geq 0$. Because we should have the condition $F_k(I_k^*) + (\ln 2)(\lambda_k^* - \lambda_{k+1}^*) = 0$ satisfied for the optimum solution we conclude that $\lambda_k^* > 0$ which using the complementary slackness condition results in $I_{k-1}^* = I_k^*$.
3. If $\alpha_k > \beta_k$ and $\mathfrak{h}_{k+1}\beta_k > \mathfrak{h}_{k-1}\alpha_k$ then we have $\tilde{I}_k > 0$. Moreover, we have $F_k(x) > 0$ for $x > \tilde{I}_k$ and $F_k(x) < 0$ for $x < \tilde{I}_k$. Now, there exists three different situations.
   - If $\tilde{I}_k > P$ we conclude that $F_k(x) < 0$ for $x \leq P$. Because we should have the condition $F_k(I_k^*) + (\ln 2)(\lambda_k^* - \lambda_{k+1}^*) = 0$ for the optimum solution we conclude that $\lambda_k^* > 0$ which by using the complementary slackness condition results in $I_k^* = I_{k-1}^*$.
   - If $I_{k+1}^* \leq \tilde{I}_k \leq P$ then we have $I_k^* = \tilde{I}_k$ and we have also $\lambda_k = \lambda_{k+1}$.
   - If $\tilde{I}_k < I_{k+1}^*$ we conclude that $F_k(I) > 0$ for $I \geq I_{k+1}^*$. Because we should have the condition $F_k(I_k^*) + (\ln 2)(\lambda_k^* - \lambda_{k+1}^*) = 0$ for the optimum solution we conclude that $\lambda_{k+1}^* > 0$ which using the complementary slackness condition results in $I_k^* = I_{k+1}^*$.

Lemma 7.5. *If we have $0 = \tilde{I}_s < \tilde{I}_{s-1} < \cdots < \tilde{I}_1 < \tilde{I}_0 = P$, then the optimal power allocation is determined by $I_k^* = \tilde{I}_k$ where $\tilde{I}_k$ is defined in* (7.103).

Proof. From the above discussions we know that $\tilde{I}_k > 0$ only if $\alpha_k > \beta_k$ and $\mathfrak{h}_{k+1}\beta_k > \mathfrak{h}_{k-1}\alpha_k$. So as mentioned before $\forall k \in [1:s-1]$ we have $F_k(x) > 0$ for $x > \tilde{I}_k$ and $F_k(x) < 0$ for $x < \tilde{I}_k$.

Now, it can be easily checked that the solution $I_k^* = \tilde{I}_k$ satisfies the set of conditions stated in (7.84) with $\lambda_k^* = 0$. Because $F_k(x) > 0$ for $x > \tilde{I}_k$ and $F_k(x) < 0$ for $x < \tilde{I}_k$ so every deviation of I_k^* from the $\tilde{I}_k$ results in a contradiction. To show this we proceed as follows.

Let us fix k. If $I_k^* > \tilde{I}_k$ then $F_k(I_k^*) > 0$ and because we have $F_k(I_k^*) + (\ln 2)(\lambda_k^* - \lambda_{k+1}^*) = 0$ then we can conclude that $\lambda_{k+1}^* > 0$. So by the slackness condition given in (7.84) we have to have $I_k^* = I_{k+1}^*$. Now, two cases may happen. First, if $I_{k+1}^* < \tilde{I}_k$ that results in a contradiction because $I_k^* > \tilde{I}_k$ and we should have $I_k^* = I_{k+1}^*$. Secondly, if $I_{k+1}^* > \tilde{I}_k > \tilde{I}_{k+1}$, then similar to the above argument we can show that $I_{k+1}^* = I_{k+2}^*$. Then we either encounter a contradiction in this step or have to continue. Finally, if we did not have any contradiction in these steps we would have $I_k^* = I_{k+1}^* = \cdots = I_s^* = 0$. Now, this is a contradiction because we had assumed $I_k^* > \tilde{I}_k > \tilde{I}_s = 0$.

So the unique solution to the set of conditions (7.84) is given by $I_k^* = \tilde{I}_k$ and $\lambda_k^* = 0$ and we are done. □

Now, by using Lemma 7.5 we may re-derive the achievable secrecy rate of (7.88) as stated in Corollary 7.1.

Corollary 7.1. *If we have high dynamic regime which means $\mathfrak{h}_i \gg \mathfrak{h}_i + 1$ then for the maximum achievable secret key generation rate we have*

$$\mathfrak{R}_s^{\mathsf{gaus}} \doteq \sum_{i=1}^{s} \Delta_i(\alpha_{i-1} - \alpha_i) \log \mathrm{SNR} = \sum_{i=1}^{s} \Delta_i \log \frac{\mathfrak{h}_i}{\mathfrak{h}_{i-1}}, \tag{7.104}$$

where $\mathfrak{h}_i = \mathrm{SNR}^{-\alpha_i}$ for $i \in [0:s]$ such that $\alpha_s < \alpha_{s-1} < \cdots < \alpha_0 < 1$.

Proof. Because $\mathfrak{h}_i \gg \mathfrak{h}_i + 1$ we have $\tilde{I}_k \doteq \frac{\Delta_{k+1}}{\Delta_k}\mathrm{SNR}^{\alpha_i}$. The conditions for Lemma 7.5 are satisfied so $I_k^* = \tilde{I}_k$ and for R_i, the secret key rate of each layer, we can write

$$\begin{aligned} R_i &= \log\left(\frac{1+\mathfrak{h}_i I_{i-1}}{1+\mathfrak{h}_i I_i} \cdot \frac{1+\mathfrak{h}_{i-1} I_i}{1+\mathfrak{h}_{i-1} I_{i-1}}\right) \\ &\doteq \log\left(\frac{1+\mathrm{SNR}^{\alpha_{i-1}-\alpha_i}\frac{\Delta_i}{\Delta_{i-1}}}{1+\frac{\Delta_{i+1}}{\Delta_i}} \cdot \frac{1+\mathrm{SNR}^{-(\alpha_{i-1}-\alpha_i)\frac{\Delta_{i+1}}{\Delta_i}}}{1+\frac{\Delta_i}{\Delta_{i-1}}}\right) \\ &\doteq (\alpha_{i-1}-\alpha_i)\log \mathrm{SNR} \\ &= \log \frac{\mathfrak{h}_i}{\mathfrak{h}_{i-1}}, \end{aligned} \tag{7.105}$$

and we are done. □

Corollary 7.2. *If $s = 2$ then based on different values of channel coefficients and probability distribution over the states we have the following cases:*

1. *if $\alpha_1 < \beta_1$ then we have $P_1^* = P$,*

2. *if* $\mathfrak{h}_2\beta_1 < \mathfrak{h}_0\alpha_1$ *then* $P_2^* = P$, *and finally*
3. *if* $\alpha_1 > \beta_1$ *and* $\mathfrak{h}_2\beta_1 > \mathfrak{h}_0\alpha_1$ *then* $P_2^* = \min(\tilde{I}_1, P)$ *where* $\tilde{I}_1$ *is the solution of equation* $F_1(\tilde{I}_1) = 0$.

Proof. For $s = 2$, the set of necessary conditions becomes as follows

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial I_1} = F_1(I_1) + (\ln 2)(\lambda_1 - \lambda_2) = 0, \\ \lambda_1[I_1 - P] = 0, \\ \lambda_2[0 - I_1] = 0, \\ \lambda_1 \geq 0, \quad \lambda_2 \geq 0. \end{cases} \tag{7.106}$$

Case 1: If $\alpha_1 < \beta_1$ then we have $\mathfrak{h}_2\beta_1 > \mathfrak{h}_0\alpha_1$ or equivalently $\tilde{I}_1 < 0$. This means $F_1(x) > 0$ for $x \geq 0$. Because, we should have $F_1(I_1^*)+(\ln 2)(\lambda_1^*-\lambda_2^*) = 0$ then we conclude that $\lambda_2^* > 0$ which by using the complementary slackness condition results in $I_1^* = 0$ or $P_2^* = 0$. So finally we have $P_1^* = P$.
Case 2: If $\mathfrak{h}_2\beta_1 < \mathfrak{h}_0\alpha_1$ then we have $\alpha_1 > \beta_1$ or equivalently $\tilde{I}_1 < 0$. This means $F_1(x) < 0$ for $x \geq 0$. Because, we should have $F_1(I_1^*)+(\ln 2)(\lambda_1^*-\lambda_2^*) = 0$ then we conclude that $\lambda_1^* > 0$ which by using the complementary slackness condition results in $I_1^* = P$ or $P_2^* = P$.
Case 3: If $\alpha_1 > \beta_1$ and $\mathfrak{h}_2\beta_1 > \mathfrak{h}_0\alpha_1$ then we have $\tilde{I}_1 > 0$. This means that $F_1(x) > 0$ for $x > \tilde{I}_1$ and $F_1(x) < 0$ for $x < \tilde{I}_1$. Then as argued in the previous cases it can be easily shown that the optimal power allocation is $P_2^* = I_1^* = \min(\tilde{I}_1, P)$. □

7.C Generalized Linear Fractional Programming (GLFP)

In this section, we state the fundamental definitions and results that are the foundation of numerical algorithms introduced in [95] to solve *Generalized Linear Fractional Programs* (GLFP) which is defined precisely in Definition 7.4 (see also [96] for the special application to wireless power control problems). It is worth mentioning that GLFPs are in general non-convex optimization problems.

With a proper change of variables, the power optimization problem derived in Section 7.7.2 can be converted to a GLFP. This enables us to find numerically the optimal power allocation for the problem of group secret key sharing over a Gaussian broadcast channel introduced in Chapter 7.

Let us start with the following definition.

Definition 7.4 ([95])**.** *We say that an optimization problem belongs to the class of Generalized Linear Fractional Programming (GLFP), if it can be represented by*

$$\begin{cases} \max & \Phi\left(\frac{f_1(\boldsymbol{x})}{g_1(\boldsymbol{x})}, \ldots, \frac{f_m(\boldsymbol{x})}{g_m(\boldsymbol{x})}\right) \\ \text{subject to} & \boldsymbol{x} \in \mathcal{D}, \end{cases} \tag{7.107}$$

or

$$\begin{cases} \min & \Phi\left(\frac{f_1(\boldsymbol{x})}{g_1(\boldsymbol{x})}, \ldots, \frac{f_m(\boldsymbol{x})}{g_m(\boldsymbol{x})}\right) \\ \text{subject to} & \boldsymbol{x} \in \mathcal{D}, \end{cases} \tag{7.108}$$

where $\mathcal{D}$ is a nonempty polytope in $\mathbb{R}^n$, and functions $f_1, \ldots, f_m$ and $g_1, \ldots, g_m$ are linear affine functions on $\mathbb{R}^n$ such that

$$-\infty < a_i \triangleq \min_{\boldsymbol{x} \in \mathcal{D}} \frac{f_i(\boldsymbol{x})}{g_i(\boldsymbol{x})} < \infty. \tag{7.109}$$

Moreover, $\Phi : \mathbb{R}^m \mapsto \mathbb{R}$ is a continuous function, increasing on $\mathbb{R}^m_{\boldsymbol{a}+} \triangleq \{\boldsymbol{y} \in \mathbb{R}^m | y_i \geq a_i, i \in [1:m]\}$, i.e., satisfying

$$\boldsymbol{a} \preceq \boldsymbol{y}' \preceq \boldsymbol{y} \qquad \Rightarrow \qquad \Phi(\boldsymbol{y}') \preceq \Phi(\boldsymbol{y}). \tag{7.110}$$

Since the problem derived in Section 7.7 is of the form (7.107), from here on we only focus on the maximization problem, namely, (7.107). As discussed in [95, Section 3], by simple manipulation we can always reduce the problem (7.107) to the case where we have

$$\min\{g_i(\boldsymbol{x}), f_i(\boldsymbol{x})\} > 0 \quad \forall \boldsymbol{x} \in \mathcal{D}, \quad i = [1:m]. \tag{7.111}$$

Also without loss of generality we may assume that $\Phi : \mathbb{R}^m_+ \to \mathbb{R}_{++}$.

Under this assumption it is possible to reformulate problem (7.107) as a monotonic optimization problem. Let us define

$$\mathcal{G} \triangleq \left\{ \boldsymbol{y} \in \mathbb{R}^m_+ \mid y_i \leq \frac{f_i(\boldsymbol{x})}{g_i(\boldsymbol{x})} \quad \forall i = [1:m], \quad \boldsymbol{x} \in \mathcal{D} \right\}. \tag{7.112}$$

Then it is possible to state the following result.

Theorem 7.13. *The optimization problem stated in 7.107 is equivalent to the following problem*

$$\begin{cases} \max & \Phi(\boldsymbol{y}) \\ \text{subject to} & \boldsymbol{y} \in \mathcal{G}. \end{cases} \tag{7.113}$$

More precisely, if $\boldsymbol{x}^*$ solves (7.107) then $\boldsymbol{y}^*$ with $y_i^* = \frac{f_i(\boldsymbol{x}^*)}{g_i(\boldsymbol{x}^*)}$ solves (7.113). Conversely, if $\boldsymbol{y}^*$ solves (7.113) and $\boldsymbol{x}^* \in \mathcal{D}$ satisfies $y_i^* \leq \frac{f_i(\boldsymbol{x}^*)}{g_i(\boldsymbol{x}^*)}$, then $\boldsymbol{x}^*$ solves (7.107).

Normal Set and Polyblock

Here we review some definitions from [95]. If $\boldsymbol{z} \in \mathbb{R}^m_+$, the hyperrectangle $[0, \boldsymbol{z}] \triangleq \{\boldsymbol{y} \in \mathbb{R}^m_+ | 0 \preceq \boldsymbol{y} \preceq \boldsymbol{z}\}$ is called a *box*. Given any finite set $\mathcal{T} \subset \mathbb{R}^m_+$ the union of all the boxes $[0, \boldsymbol{z}]$, $\boldsymbol{z} \in \mathcal{T}$, is called a *polyblock* with vertex set $\mathcal{T}$. A vertex $\boldsymbol{z} \in \mathcal{T}$ is said to be *proper* if $\boldsymbol{z}$ is not dominated by any other vertex, i.e., if there is no $\boldsymbol{z}' \in \mathcal{T}$ such that $\boldsymbol{z}' \neq \boldsymbol{z}$ and $\boldsymbol{z}' \succeq \boldsymbol{z}$. Obviously a polyblock is fully determined by its proper vertices.

A set $\mathcal{H} \subset \mathbb{R}^m_+$ is called *normal* if $\boldsymbol{y} \in \mathcal{H}$ always implies that $[0, \boldsymbol{y}] \subset \mathcal{H}$. A polyblock, in particular a box, is normal. The orthant $\mathbb{R}^m_+$ and the empty set are also normal sets. The intersection of any family of normal sets is obviously a normal set. The intersection of finitely many polyblocks is a polyblock. If

$\mathcal{D} \subset \mathbb{R}^n$ and $w : \mathcal{D} \to \mathbb{R}_+^m$ is any nonnegative-valued function on $\mathcal{D}$ then the set $\mathcal{H} = \{\boldsymbol{y} \in \mathbb{R}_+^m \mid \boldsymbol{y} \preceq w(\boldsymbol{x}),\ \boldsymbol{x} \in \mathcal{D}\}$ is normal.

A point $\boldsymbol{y} \in \mathbb{R}_+^m$ is called an *upper boundary point* of a nonempty normal set $\mathcal{H} \subset \mathbb{R}_+^m$ if $\alpha\boldsymbol{y} \in \mathcal{H}$, $\forall \alpha < 1$ but $\alpha\boldsymbol{y} \notin \mathcal{H}$, $\forall \alpha > 1$. The set of upper boundary points of $\mathcal{H}$ is called the *upper boundary* of $\mathcal{H}$ and is denoted by $\partial^+\mathcal{H}$. Then [95, Proposition 2] states the following result:

Proposition 7.1. *Let $\Phi(\boldsymbol{y}) : \mathbb{R}_+^m \to \mathbb{R}$ be an increasing function. The maximum of $\Phi(\boldsymbol{y})$ over a polyblock is attained at one proper vertex of this polyblock. The maximum of $\Phi(\boldsymbol{y})$ over a nonempty compact normal set $\mathcal{H}$ is attained on $\partial^+\mathcal{H}$.*

It can be easily observed that the set $\mathcal{G}$ defined in (7.112) is normal so the maximum value of the problem (7.113) is attained on the upper boundary of $\mathcal{G}$. To find the optimal value of this problem we can use [95, Algorithm 1]. The core idea of this algorithm is based on approximating $\partial^+\mathcal{G}$ by a polyblock in every iteration of the algorithm and then improving this approximation in each iteration until finding a good estimate of the global optimum value of the problem (7.113). It is shown in [95, Theorem] that by tolerating a non-zero error value in the solution, the proposed algorithm converges in finite time to an estimation of the global optimum value.

7.D Rewriting the Power Allocation Problem as a GLFP

For convenience, let us rewrite the power allocation optimization problem (7.81) in the following

$$R_s = \begin{cases} \max & \sum_{i=1}^{s} \Delta_i R_i \\ \text{subject to} & I_k \leq I_{k-1},\ \forall k \in [1:s], \end{cases} \tag{7.114}$$

where $I_0 = P$, $I_s = 0$, and we have

$$R_i = \log\left(\frac{1+h_i^2 I_{i-1}}{1+h_i^2 I_i} \cdot \frac{1+h_{i-1}^2 I_i}{1+h_{i-1}^2 I_{i-1}}\right). \tag{7.115}$$

Note that we can write

$$\sum_{i=1}^{s} \Delta_i R_i = \log\left[\prod_{i=1}^{s}\left(\frac{1+h_i^2 I_{i-1}}{1+h_i^2 I_i}\right)^{\Delta_i}\left(\frac{1+h_{i-1}^2 I_i}{1+h_{i-1}^2 I_{i-1}}\right)^{\Delta_i}\right]. \tag{7.116}$$

Now by defining

$$\Phi(y_1, \ldots, y_{2s}) = \log\left[\prod_{i=1}^{s} y_{2i-1}^{\Delta_i} \cdot y_{2i}^{\Delta_i}\right], \tag{7.117}$$

which is an increasing function in $\boldsymbol{y}$, and by defining

$$y_{2i-1} = \frac{1 + h_i^2 I_{i-1}}{1 + h_i^2 I_i}, \quad i \in [1 : s], \tag{7.118}$$

and

$$y_{2i} = \frac{1 + h_{i-1}^2 I_i}{1 + h_{i-1}^2 I_{i-1}}, \quad i \in [1 : s], \tag{7.119}$$

it can be easily observed that (7.114) is a GLFP of the form (7.107). So the numerical methods developed in [95, 96] can be used to solve the power allocation problem (7.114).

"Myth is the hidden part of every story, the buried part, the region that is still unexplored because there are as yet no words to enable us to get there. Myth is nourished by silence as well as by words."

- Italo Calvino

Group Secret Key Agreement in a Linear Non-coherent Packetized Networks

8

For communication over a network performing linear NC, Cai and Yeung [98] introduced the problem of securing a multicast transmission against an eavesdropper. In particular, consider a network implementing linear NC over a finite field $\mathbb{F}_q$. Let us assume that the min-cut value from the source to each receiver is c. From the main theorem of NC (see Theorem 2.1) [6, 5], we know that a source can send information at rate equal to the min-cut c to the destinations, in the absence of any malicious eavesdropper. Now, suppose there is a passive eavesdropper, Eve, who overhears ρ arbitrary edges in the network. The *secure NC* problem is to design a coding scheme such that Eve does not obtain any information about the messages transmitted from the source to destinations. Cai and Yeung [98] showed that the secrecy capacity for this problem is $c - \rho$ and can be achieved if the field size q is sufficiently large. Later this problem formulation has been investigated in many other works. Feldman et al. [99] showed that by sacrificing a small amount of rate, one might find a secure scheme that requires much smaller field size. Rouayheb et al. [100] observed that this problem can be considered as a generalization of the Ozarow-Wyner wiretap channel of type II. Silva et al. [73] proposed a universal coding scheme that only employs encoding at the source.

In contrast to the previous work, in this chapter we study the problem of secret key sharing among multiple terminals when nodes can send feedback over a public channel. We consider a source multicasting information over a network at rate equal to the min-cut c to the destinations. We also assume that the relay nodes in the network perform randomized linear NC which is modeled by a non-coherent transmission scheme. Motivated by [33, 26], we model a non-coherent NC scenario by a multiplicative matrix channel over a finite field $\mathbb{F}_q$ with uniform and i.i.d. distribution over transfer matrices in every time-slot.

The problem of key agreement between a set of terminals with access to noisy broadcast channel and public discussion channel (visible to the eavesdropper) was studied in [85], where some achievable secrecy rates were established, assuming Eve does not have access to the noisy broadcast transmissions. This was generalized in [92, 86] by developing (non-computable) outer bounds for secrecy rates. However, to the best of our knowledge, ours is the first work to consider multi-terminal secret key agreement over networks employing randomized NC, when a passive eavesdropper has access to the broadcast transmissions.

Our contributions in this chapter are as follows. For the secret key sharing problem introduced above, we propose an asymptotic achievability scheme assuming that the field size q is large. This scheme is based on *subspace coding* and can be extended for arbitrary number of terminals. Using the result of [85], we derive an upper bound for this problem. For $m = 1$, the proposed lower bound matches the upper bound and the *secret key generation capacity* is characterized. However, for $m \geq 2$, depending on the channel parameters, the upper and lower bound might match or not.

8.1 Problem Statement

We consider a set of $m + 1 \geq 2$ honest nodes, $\mathsf{T}_0, \ldots, \mathsf{T}_m$, ($\mathsf{T}$ stands for "terminal") that aim to share a secret key $\mathcal{K}$ among themselves while keeping it concealed from a passive adversary, Eve. Eve does not perform any transmissions, but is trying to eavesdrop on (overhear) the communications between the honest nodes. For convenience, sometimes we will refer to node $\mathsf{T}_0, \mathsf{T}_1, \mathsf{T}_2, \ldots$, as "Alice," "Bob," "Calvin," and so on.

We assume that there exists a non-coherent NC broadcast channel (which is going to be defined more precisely in the following) from Alice to the other terminals (including Eve). Also we assume that the legitimate terminals can publicly discuss over a noiseless rate unlimited public channel.

Consider a non-coherent linear NC communication scenario where at every time-slot t Alice injects a set of n_{A} vectors (packets) of length L (over some finite field $\mathbb{F}_q$) into the network, denoted by the row vectors of the matrix $\boldsymbol{X}_{\mathsf{A}}[t] \in \mathbb{F}_q^{n_A \times L}$. Each terminal T_i receives n_i randomly chosen linear combinations of the transmitted vectors, namely for $r \in \{\mathsf{T}_1, \ldots, \mathsf{T}_m, \mathsf{E}\}$, we have [1]

$$\boldsymbol{X}_r[t] = \boldsymbol{F}_r[t]\boldsymbol{X}_{\mathsf{A}}[t], \tag{8.1}$$

where $\boldsymbol{F}_r[t] \in \mathbb{F}_q^{n_r \times n_{\mathsf{A}}}$ is chosen uniformly at random among all possible matrices and independently for each receiver and every time-slot. So for the channel transition probability we can write

$$P_{\boldsymbol{X}_1 \cdots \boldsymbol{X}_m \boldsymbol{X}_{\mathsf{E}} | \boldsymbol{X}_{\mathsf{A}}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m, \boldsymbol{x}_{\mathsf{E}} | \boldsymbol{x}_{\mathsf{A}}) = P_{\boldsymbol{X}_{\mathsf{E}} | \boldsymbol{X}_{\mathsf{A}}}(\boldsymbol{x}_{\mathsf{E}} | \boldsymbol{x}_{\mathsf{A}}) \prod_{i=1}^{m} P_{\boldsymbol{X}_i | \boldsymbol{X}_{\mathsf{A}}}(\boldsymbol{x}_i | \boldsymbol{x}_{\mathsf{A}}), \tag{8.2}$$

1. During the chapter, we use T_i and i interchangeably when they are used as subscript. So instead of $\boldsymbol{X}_{\mathsf{T}_i}$ we sometimes write $\boldsymbol{X}_i$. At some points, we also use $\boldsymbol{X}_{\mathsf{A}}$, $\boldsymbol{X}_{\mathsf{B}}$, $\boldsymbol{X}_{\mathsf{C}}$, etc., to denote for $\boldsymbol{X}_0$, $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, etc.

where for each $r \in \{\mathsf{T}_1, \ldots, \mathsf{T}_m, \mathsf{E}\}$ we have (see Section 3.3.1)

$$P_{\boldsymbol{X}_r|\boldsymbol{X}_{\mathsf{A}}}(\boldsymbol{x}_r|\boldsymbol{x}_{\mathsf{A}}) \triangleq \begin{cases} q^{-n\dim(\boldsymbol{x}_{\mathsf{A}})} & \text{if } \langle \boldsymbol{x}_r \rangle \sqsubseteq \langle \boldsymbol{x}_{\mathsf{A}} \rangle, \\ 0 & \text{otherwise.} \end{cases} \tag{8.3}$$

Note that in this setup we do not assume any CSI [2] at the transmitter or receivers.

Here, we define the secret key sharing capacity in the same way that we have defined it in Chapter 7. So we do not repeat Definitions 7.1 and 7.2 again.

8.2 Main Results

In this section, we state the main result of this chapter which is upper and lower bounds on the secret key sharing capacity $C_s^{\mathsf{non-coh}}$ among multiple terminals having access to a non-coherent NC channel as defined in Section 8.1. This result is stated in the following theorem.

Theorem 8.1. *The secret key generation capacity among $m+1$ terminals, as defined in Section 7.2, that have access to a non-coherent multicast NC channel defined in Section 8.1, is upper bounded by*

$$C_s^{\mathsf{non-coh}} \leq \min_{i\in[1:m]} \Big[\big(\min[n_{\mathsf{A}}, n_i + n_{\mathsf{E}}] - n_{\mathsf{E}} \big) \big(L - \min[n_{\mathsf{A}}, n_i + n_{\mathsf{E}}] \big) \Big] \log q. \tag{8.4}$$

Moreover, there exists an efficient achievability scheme that can achieve the secrecy rates less than the solution to the following convex optimization problem

$$\begin{array}{ll} \text{maximize} & \big[\min_{r\in[1:m]} \sum_{\mathcal{J}\ni r} \theta_{\mathcal{J}}\big] (L - n_{\mathsf{A}}) \log q \\ \text{subject to} & \theta_{\mathcal{J}} \geq 0, \quad \forall \mathcal{J} \subseteq [1:m],\ \mathcal{J} \neq \emptyset, \quad \text{and} \\ & \theta_{\mathcal{J}_1} + \cdots + \theta_{\mathcal{J}_k} \leq \\ & \qquad \dim\left(U_{\mathcal{J}_1} + \cdots + U_{\mathcal{J}_k} + \Pi_{\mathsf{E}}\right) - \dim(\Pi_{\mathsf{E}}) \\ & \forall k,\ \forall \mathcal{J}_1, \ldots, \mathcal{J}_k :\ \emptyset \neq \mathcal{J}_i \subseteq [1:m], \\ & \mathcal{J}_i \neq \mathcal{J}_j \text{ if } i \neq j, \end{array} \tag{8.5}$$

where for every $\mathcal{J}$, $U_{\mathcal{J}}$ is chosen uniformly at random from $\Pi_{\mathcal{J}}$ with the dimension calculated by (8.15) *under the assumption that $\Pi_1, \ldots, \Pi_m$, and Π_{E} are selected independently and uniformly at random from Π_{A} with dimensions $n_1, \ldots, n_m$, and n_{E} respectively.*

The stated upper bound is proved in Section 8.3. The proof of the lower bound can also be found in Section 8.4.

2. Channel state information.

8.3 Upper Bound for Non-coherent NC Channel

In Section 7.4, we have shown that the secret key generation rate among multiple terminals having access to the output of an independent broadcast channel can be upper bounded by (7.26), i.e.,

$$C_s \leq \min_{j\in[1:m]} \max_{P_{\boldsymbol{X}_\mathsf{A}}} I(\boldsymbol{X}_\mathsf{A};\boldsymbol{X}_j|\boldsymbol{X}_\mathsf{E}). \tag{8.6}$$

Now, we need to evaluate the above upper bound for the non-coherent NC channel defined in Section 8.1. To this end we proceed as follows. First, we state the following lemma.

Lemma 8.1. *For the joint distribution of the form*

$$P_{\boldsymbol{X}_\mathsf{A}\boldsymbol{X}_i\boldsymbol{X}_\mathsf{E}}(\boldsymbol{x}_\mathsf{A},\boldsymbol{x}_i,\boldsymbol{x}_\mathsf{E}) = P_{\boldsymbol{X}_\mathsf{A}}(\boldsymbol{x}_\mathsf{A})P_{\boldsymbol{X}_i|\boldsymbol{X}_\mathsf{A}}(\boldsymbol{x}_i|\boldsymbol{x}_\mathsf{A})P_{\boldsymbol{X}_\mathsf{E}|\boldsymbol{X}_\mathsf{A}}(\boldsymbol{x}_\mathsf{E}|\boldsymbol{x}_\mathsf{A}) \tag{8.7}$$

the mutual information $I(\boldsymbol{X}_\mathsf{A};\boldsymbol{X}_i|\boldsymbol{X}_\mathsf{E})$ is a concave function of $P_{\boldsymbol{X}_\mathsf{A}}(\boldsymbol{x}_\mathsf{A})$ for fixed $P_{\boldsymbol{X}_i|\boldsymbol{X}_\mathsf{A}}(\boldsymbol{x}_i|\boldsymbol{x}_\mathsf{A})$ and $P_{\boldsymbol{X}_\mathsf{E}|\boldsymbol{X}_\mathsf{A}}(\boldsymbol{x}_\mathsf{E}|\boldsymbol{x}_\mathsf{A})$.

Proof. For the proof refer to Appendix 8.B. □

Similar to Definition 3.2, here we define an equivalent subspace broadcast channel from Alice to the rest of terminals as follows. We assume that Alice sends a subspace $\Pi_\mathsf{A} \in \mathrm{Sp}(L, n_\mathsf{A})$ where $\Pi_\mathsf{A} = \langle \boldsymbol{X}_\mathsf{A} \rangle$ and each of the legitimate terminals receives $\Pi_i \in \mathrm{Sp}(L, n_i)$ and Eve receives $\Pi_\mathsf{E} \in \mathrm{Sp}(L, n_\mathsf{E})$ where $\Pi_i = \langle \boldsymbol{X}_i \rangle$ and $\Pi_\mathsf{E} = \langle \boldsymbol{X}_\mathsf{E} \rangle$, respectively. Recall that the set $\mathrm{Sp}(L,k)$ is defined in Definition 2.3. The channel transition probabilities are independent and for each receiver i is defined as follows

$$P_{\Pi_i|\Pi_\mathsf{A}}(\pi_i|\pi_\mathsf{A}) \triangleq \begin{cases} \psi(n_i, \dim(\pi_i)) q^{-n_i \dim(\pi_\mathsf{A})} & \text{if } \pi_i \sqsubseteq \pi_\mathsf{A}, \\ 0 & \text{otherwise,} \end{cases} \tag{8.8}$$

where function ψ is defined in Definition 2.4 (see also Lemma 2.5 and Remark 2.1).

Lemma 8.2. *For every input distribution $P_{\boldsymbol{X}_\mathsf{A}}$ there exists an input distribution P_{Π_A} such that $I(\boldsymbol{X}_\mathsf{A};\boldsymbol{X}_i|\boldsymbol{X}_\mathsf{E}) = I(\Pi_\mathsf{A};\Pi_i|\Pi_\mathsf{E})$ and vice-versa.*

Proof. We can expand $I(\boldsymbol{X}_\mathsf{A};\boldsymbol{X}_i|\boldsymbol{X}_\mathsf{E}) = I(\boldsymbol{X}_\mathsf{A};\boldsymbol{X}_i\boldsymbol{X}_\mathsf{E}) - I(\boldsymbol{X}_\mathsf{A};\boldsymbol{X}_\mathsf{E})$. Using Theorem 3.1, by defining $\Pi_\mathsf{A} = \langle \boldsymbol{X}_\mathsf{A} \rangle$ and $\Pi_r = \langle \boldsymbol{X}_r \rangle$ for $r \in \{\mathsf{T}_1, \ldots, \mathsf{T}_m, \mathsf{T}_\mathsf{E}\}$, we can write

$$\begin{aligned} I(\boldsymbol{X}_\mathsf{A};\boldsymbol{X}_i|\boldsymbol{X}_\mathsf{E}) &= I(\boldsymbol{X}_\mathsf{A};\boldsymbol{X}_i,\boldsymbol{X}_\mathsf{E}) - I(\boldsymbol{X}_\mathsf{A};\boldsymbol{X}_\mathsf{E}) \\ &\overset{(a)}{=} I(\Pi_\mathsf{A};\Pi_i+\Pi_\mathsf{E}) - I(\Pi_\mathsf{A};\Pi_\mathsf{E}) \\ &\overset{(b)}{\leq} I(\Pi_\mathsf{A};\Pi_i,\Pi_\mathsf{E}) - I(\Pi_\mathsf{A};\Pi_\mathsf{E}) \\ &= I(\Pi_\mathsf{A};\Pi_i|\Pi_\mathsf{E}) \end{aligned} \tag{8.9}$$

where (a) follows from Theorem 3.1 and (b) is true because of the data processing inequality applied on the Markov chain $\Pi_i + \Pi_{\mathsf{E}} \leftrightarrow (\Pi_i, \Pi_{\mathsf{E}}) \leftrightarrow \Pi_{\mathsf{A}}$. On the other hand, by applying data processing inequality for another time, we can write

$$\begin{aligned} I(\boldsymbol{X}_{\mathsf{A}}; \boldsymbol{X}_i|\boldsymbol{X}_{\mathsf{E}}) &= I(\boldsymbol{X}_{\mathsf{A}}; \boldsymbol{X}_i, \boldsymbol{X}_{\mathsf{E}}) - I(\boldsymbol{X}_{\mathsf{A}}; \boldsymbol{X}_{\mathsf{E}}) \\ &\overset{\text{(a)}}{\geq} I(\Pi_{\mathsf{A}}; \Pi_i, \Pi_{\mathsf{E}}) - I(\boldsymbol{X}_{\mathsf{A}}; \boldsymbol{X}_{\mathsf{E}}) \\ &\overset{\text{(b)}}{=} I(\Pi_{\mathsf{A}}; \Pi_i, \Pi_{\mathsf{E}}) - I(\Pi_{\mathsf{A}}; \Pi_{\mathsf{E}}) \\ &= I(\Pi_{\mathsf{A}}; \Pi_i|\Pi_{\mathsf{E}}) \end{aligned} \tag{8.10}$$

where (a) is true because of the Markov chain $(\Pi_i, \Pi_{\mathsf{E}}) \leftrightarrow (\boldsymbol{X}_i, \boldsymbol{X}_{\mathsf{E}}) \leftrightarrow \boldsymbol{X}_{\mathsf{A}} \leftrightarrow \Pi_{\mathsf{A}}$ and (b) is true because of Theorem 3.1. Hence we are done. □

So by Lemma 8.2, in order to maximize $I(\boldsymbol{X}_{\mathsf{A}}; \boldsymbol{X}_i|\boldsymbol{X}_{\mathsf{E}})$ with respect to $P_{\boldsymbol{X}_{\mathsf{A}}}$ it is sufficient to solve an equivalent problem, i.e., maximize $I(\Pi_{\mathsf{A}}; \Pi_i|\Pi_{\mathsf{E}})$ with respect to $P_{\Pi_{\mathsf{A}}}$; which is seemingly a simpler optimization problem than the original one.

Lemma 8.3. *The input distribution that maximizes $I(\Pi_{\mathsf{A}}; \Pi_i|\Pi_{\mathsf{E}})$ is the one which is uniform over all subspaces having the same dimension.*

Proof. By the concavity of $I(\Pi_{\mathsf{A}}; \Pi_i|\Pi_{\mathsf{E}})$ with respect to $P_{\Pi_{\mathsf{A}}}$, which is stated in Lemma 8.1, the proof follows by an argument very similar to the proof of Lemma 3.4 (see Appendix 3.A). □

Lemma 8.4. *Asymptotically in the field size we have*

$$\begin{aligned} \max_{P_{\boldsymbol{X}_{\mathsf{A}}}} I(\boldsymbol{X}_{\mathsf{A}}; \boldsymbol{X}_i|\boldsymbol{X}_{\mathsf{E}}) &= \max_{P_{\Pi_{\mathsf{A}}}} I(\Pi_{\mathsf{A}}; \Pi_i|\Pi_{\mathsf{E}}) \\ &= \big(\min[n_{\mathsf{A}}, n_i + n_{\mathsf{E}}] - n_{\mathsf{E}}\big)\big(L - \min[n_{\mathsf{A}}, n_i + n_{\mathsf{E}}]\big) \log q. \end{aligned} \tag{8.11}$$

Proof. For the proof refer to Appendix 8.B. □

Thus, by using the upper bound given in (7.26) and Lemma 8.4 we have the following result for the upper bound on the secret key generation rate, as stated in Theorem 8.2.

Theorem 8.2. *The secret key generation rate in a non-coherent NC scenario, which is defined in Section 8.1, is upper bounded by*

$$\begin{aligned} &C_s^{\mathsf{non-coh}} \leq \\ &\min_{i \in [1:m]} \Big[\big(\min[n_{\mathsf{A}}, n_i + n_{\mathsf{E}}] - n_{\mathsf{E}}\big)\big(L - \min[n_{\mathsf{A}}, n_i + n_{\mathsf{E}}]\big)\Big] \log q. \end{aligned} \tag{8.12}$$

Remark 8.1. *Note that if $n_{\mathsf{E}} = n_{\mathsf{A}}$ then the secret key generation rate is zero because Eve is so powerful that she overhears all of the transmitted information.*

8.4 Asymptotic Achievability Scheme

Here in this section, we describe our achievability scheme for the secret key sharing problem among multiple terminals in a non-coherent NC setup.

Without loss of generality, let us assume that [3] $n_{\mathsf{A}} < L$. Moreover, here we only focus an the asymptotic regime where the field size is large. Suppose that Alice broadcasts a message $\boldsymbol{X}_{\mathsf{A}}[t]$ at time-slot t of the following form

$$\boldsymbol{X}_{\mathsf{A}}[t] = \left[\begin{array}{cc} \boldsymbol{I}_{n_{\mathsf{A}} \times n_{\mathsf{A}}} & \boldsymbol{M}[t] \end{array} \right], \tag{8.13}$$

where $\boldsymbol{M}[t] \in \mathbb{F}_q^{n_{\mathsf{A}} \times (L-n_{\mathsf{A}})}$ is a uniformly at random distributed matrix. The rest of legitimate terminals and Eve receive a linear transformed version of $\boldsymbol{X}_{\mathsf{A}}[t]$ according to the channel introduced in (8.1).

For each terminal $r \in \{\mathsf{T}_0, \ldots, \mathsf{T}_m, \mathsf{T}_{\mathsf{E}}\}$, we define the subspace $\Pi_r \triangleq \langle \boldsymbol{X}_r \rangle$. Then, for every $r \neq \mathsf{T}_0$ we have $\Pi_r \sqsubseteq \Pi_{\mathsf{A}}$. Because of (8.13), after broadcasting $\boldsymbol{X}_{\mathsf{A}}[t]$, the legitimate terminals learn the channel state and reveal the channel transfer matrices $\boldsymbol{F}_r[t]$, $r \in [1:m]$, publicly over the public channel. Thus Alice can also recover the subspaces Π_r for all of the legitimate terminals.

Now, for each non-empty subset $\mathcal{J} \subseteq [1:m]$ of legitimate receivers, let us define the subspace $U_{\mathcal{J}}$ as follows

$$U_{\mathcal{J}} \triangleq \Pi_{\mathcal{J}} \setminus_s \left(\sum_{i \in \mathcal{J}^c} \Pi_{i\mathcal{J}} + \Pi_{\mathsf{E}\mathcal{J}} \right), \tag{8.14}$$

where $\Pi_{\mathcal{J}} = \cap_{j \in \mathcal{J}} \Pi_j$, $\Pi_{i\mathcal{J}} = \Pi_i \cap \Pi_{\mathcal{J}}$, and $\Pi_{\mathsf{E}\mathcal{J}} = \Pi_{\mathsf{E}} \cap \Pi_{\mathcal{J}}$. By definition, $U_{\mathcal{J}}$ is the common subspace among the receivers in $\mathcal{J}$ which is orthogonal to all of the subspaces of other terminals, i.e., it is orthogonal to Π_i, $i \in \mathcal{J}^c$, and Π_{E} (see also Figure 8.1). Note that the subspaces $U_{\mathcal{J}}$'s are not uniquely defined. However, from the definition of the operator "$\setminus_s$", it can be easily shown that the dimension of each $U_{\mathcal{J}}$ is uniquely determined and equal to

$$\dim(U_{\mathcal{J}}) = \dim(\Pi_{\mathcal{J}}) - \dim \left(\sum_{i \in \mathcal{J}^c} \Pi_{i\mathcal{J}} + \Pi_{\mathsf{E}\mathcal{J}} \right). \tag{8.15}$$

If Alice had the subspace Π_{E} observed by Eve, she would be able to construct subspaces $U_{\mathcal{J}}$'s; but she does not have Π_{E}. However, because the subspaces Π_i's and Π_{E} are chosen independently and uniformly at random from Π_{A}, and because the field size q is large, Alice, by applying Lemma 2.11, can find the dimension of each $U_{\mathcal{J}}$ w.h.p. Then, by applying Lemma 2.10, it can be easily observed that if Alice chooses a uniformly at random subspace of $\Pi_{\mathcal{J}}$ with dimension $\dim(U_{\mathcal{J}})$ then it satisfies (8.14) w.h.p., so it can be a possible candidate for $U_{\mathcal{J}}$.

3. If $L \leq n_{\mathsf{A}}$ then Alice can reduce the number of injected packets into the network from n_{A} to some smaller number n'_{A} where $n'_{\mathsf{A}} < L$.

Now, consider $2^m - 1$ different non-empty subsets of $[1 : m]$. To each subset $\emptyset \neq \mathcal{J} \subseteq [1 : m]$, we assign a parameter $\theta_{\mathcal{J}} \geq 0$ such that the following set of inequalities hold,

$$\theta_{\mathcal{J}_1} + \cdots + \theta_{\mathcal{J}_k} \leq \dim\left(U_{\mathcal{J}_1} + \cdots + U_{\mathcal{J}_k} + \Pi_{\mathsf{E}}\right) - \dim(\Pi_{\mathsf{E}}), \tag{8.16}$$

for any $k \in [1 : 2^m - 1]$ and any different selection of subsets $\mathcal{J}_1, \ldots, \mathcal{J}_k$. Note that the right hand side of the inequalities defined in (8.16) depend on the actual choice of subspaces $U_{\mathcal{J}}$'s. But, as described above, in the following we assume that $U_{\mathcal{J}}$'s are chosen uniformly at random from $\Pi_{\mathcal{J}}$.

If Alice knows the subspace Π_{E}, then we can state the following result.

Lemma 8.5. *There exists subspaces $U'_{\mathcal{J}} \sqsubseteq U_{\mathcal{J}}$ such that $\dim(U'_{\mathcal{J}}) = \theta_{\mathcal{J}}$ for all $\emptyset \neq \mathcal{J} \subseteq [1 : m]$, and $U'_{\mathcal{J}}$'s and Π_{E} are orthogonal subspaces (i.e., $\dim(\Pi_{\mathsf{E}} + \sum_i U'_{\mathcal{J}_i}) = \dim(\Pi_{\mathsf{E}}) + \sum_i \theta_{\mathcal{J}_i}$) if and only if $\theta_{\mathcal{J}}$'s are non-negative integers and satisfy* (8.16).

Proof. The proof of this lemma is based on [101, Lemma 4] and can be found in Appendix 8.B. □

Figure 8.1 depicts pictorially the relation between subspaces introduced in the above discussions.

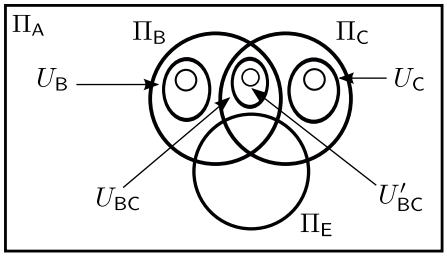


Figure 8.1 – The relations between subspaces Π's, U's, and U''s for the case of $m = 2$.

Although in practice Alice only knows the dimension of Π_{E} (w.h.p.), but still she can find subspaces $U'_{\mathcal{J}} \sqsubseteq U_{\mathcal{J}}$ such that the result of Lemma 8.5 holds w.h.p., as stated in Lemma 8.6.

Lemma 8.6. *Alice can find subspaces $U'_{\mathcal{J}} \sqsubseteq U_{\mathcal{J}}$ such that $\dim(U'_{\mathcal{J}}) = \theta_{\mathcal{J}}$ for all $\emptyset \neq \mathcal{J} \subseteq [1 : m]$, and $U'_{\mathcal{J}}$'s are orthogonal subspaces and $U'_{\mathcal{J}}$'s and Π_{E} are orthogonal subspaces w.h.p., if and only if $\theta_{\mathcal{J}}$'s are non-negative integers and satisfy* (8.16).

Proof. For the proof refer to Appendix 8.B. □

Then, we have the following result.

Theorem 8.3. *The secret key sharing rate given by the solution of the following convex optimization problem can be achieved*

$$\begin{array}{ll} \text{maximize} & \left[\min_{r\in[1:m]} \sum_{\mathcal{J}\ni r} \theta_{\mathcal{J}}\right] (L-n_{\mathsf{A}}) \log q \\ \text{subject to} & \theta_{\mathcal{J}} \geq 0, \quad \forall \mathcal{J} \subseteq [1:m],\ \mathcal{J} \neq \emptyset, \quad \text{and} \\ & \theta_{\mathcal{J}_1} + \cdots + \theta_{\mathcal{J}_k} \leq \\ & \qquad \dim\left(U_{\mathcal{J}_1} + \cdots + U_{\mathcal{J}_k} + \Pi_{\mathsf{E}}\right) - \dim(\Pi_{\mathsf{E}}) \\ & \forall k,\ \forall \mathcal{J}_1, \ldots, \mathcal{J}_k :\ \emptyset \neq \mathcal{J}_i \subseteq [1:m], \\ & \mathcal{J}_i \neq \mathcal{J}_j \text{ if } i \neq j, \end{array} \tag{8.17}$$

where for every $\mathcal{J}$, $U_{\mathcal{J}}$ is chosen uniformly at random from $\Pi_{\mathcal{J}}$ with the dimension calculated by (8.15) *under the assumption that $\Pi_1, \ldots, \Pi_m$, and Π_{E} are selected independently and uniformly at random from Π_{A} with dimensions $n_1, \ldots, n_m$, and n_{E} respectively.*

Proof. Let Alice use the broadcast channel N times by sending matrices

$$\boldsymbol{X}_{\mathsf{A}}[1], \ldots, \boldsymbol{X}_{\mathsf{A}}[N], \tag{8.18}$$

of the form (8.13). As mentioned before, in every time-slot t, each of the legitimate terminals sends publicly the channel transfer matrix it has received.

Then, let us define $\hat{\theta}_{\mathcal{J}} \triangleq \lfloor N\theta_{\mathcal{J}} \rfloor$ for all $\mathcal{J}$ and consider the following set of inequalities

$$\begin{aligned} &\hat{\theta}_{\mathcal{J}_1} + \cdots + \hat{\theta}_{\mathcal{J}_k} + N \dim(\Pi_{\mathsf{E}}) \leq \\ &\dim\left(\bigoplus_{t=1}^{N} U_{\mathcal{J}_1}[t] + \cdots + \bigoplus_{t=1}^{N} U_{\mathcal{J}_k}[t] + \bigoplus_{t=1}^{N} \Pi_{\mathsf{E}}[t]\right), \end{aligned} \tag{8.19}$$

where "$\oplus$" is the direct sum operator. Each of $\hat{U}_{\mathcal{J}_i} \triangleq \bigoplus_{t=1}^{N} U_{\mathcal{J}_i}[t]$ is a subspace of an $N \times n_{\mathsf{A}}$ dimensional space $\bigoplus_{t=1}^{N} \Pi_{\mathsf{A}}[t]$. Similarly, we have $\hat{\Pi}_{\mathsf{E}} \sqsubseteq \bigoplus_{t=1}^{N} \Pi_{\mathsf{A}}[t]$ where $\hat{\Pi}_{\mathsf{E}} \triangleq \bigoplus_{t=1}^{N} \Pi_{\mathsf{E}}[t]$. It can be easily seen that if the set of inequalities (8.16) are satisfied then the set of inequalities (8.19) are also satisfied.

Now, by using Lemma 8.6, Alice can find a set of orthogonal subspaces $\hat{U}'_{\mathcal{J}}$ with dimension $\hat{\theta}_{\mathcal{J}}$ (that are also orthogonal to $\hat{\Pi}_{\mathsf{E}}$ w.h.p.). By applying Lemma 8.7, one would observe that if Alice uses a basis of $\hat{U}'_{\mathcal{J}}$ ($\hat{\theta}_{\mathcal{J}}$ linear independent vectors from $\hat{U}'_{\mathcal{J}}$) to share a secret key $\mathcal{K}_{\mathcal{J}}$ with all terminals in $\mathcal{J}$, then this key is secure from Eve and all other legitimate terminals in $\mathcal{J}^c$ w.h.p.Using each key $\mathcal{K}_{\mathcal{J}}$, Alice can send a message of size $\hat{\theta}_{\mathcal{J}}(L-n_{\mathsf{A}}) \log q$ secretly to the terminals in $\mathcal{J}$. In order to share the key $\mathcal{K}_{\mathcal{J}}$, Alice sends publicly a set of coefficients for each terminal in $\mathcal{J}$ so that each of them can construct the subspace $\hat{U}_{\mathcal{J}}$ from their own received subspace. Note that even having these coefficients, Eve cannot recover any information regarding $\mathcal{K}_{\mathcal{J}}$ (for more discussion see proof of Theorem 7.7 in Chapter 7).

Up until now, the problem of sharing a key $\mathcal{K}$ among legitimate terminals have been reduced to a multicast problem where Alice would like to transmit a message (i.e., the shared key $\mathcal{K}$) to a set of terminal where the rth one has a

min-cut $\sum_{\mathcal{J}\ni r}\hat{\theta}_{\mathcal{J}}$. From the main theorem of NC (e.g., see [6, 5, 7, 8]), we know that this problem can be solved by performing linear NC where the achievable rate is as follows

$$\mathfrak{R}_s^{\mathsf{non-coh}} \leq \left[\frac{1}{N}\min_{r\in[1:m]}\sum_{\mathcal{J}\ni r}\hat{\theta}_{\mathcal{J}}\right](L-n_{\mathsf{A}})\log q. \tag{8.20}$$

By increasing N, the achievable secrecy rate will be arbitrarily close to

$$\mathfrak{R}_s^{\mathsf{non-coh}} \leq \left[\min_{r\in[1:m]}\sum_{\mathcal{J}\ni r}\theta_{\mathcal{J}}\right](L-n_{\mathsf{A}})\log q, \tag{8.21}$$

and we are done. □

Lemma 8.7. *Consider a set of n_{A} packets denoted by the rows of a matrix $\boldsymbol{X}_{\mathsf{A}} \in \mathbb{F}_q^{n_{\mathsf{A}}\times L}$ of the form $\boldsymbol{X}_{\mathsf{A}} = [\boldsymbol{I} \quad \boldsymbol{M}]$, where $\boldsymbol{M} \sim \mathsf{Uni}\left(\mathbb{F}_q^{n_{\mathsf{A}}\times(L-n_{\mathsf{A}})}\right)$. Assume that Eve has overheard n_{E} independent linear combinations of these packets, represented by the rows of a matrix $\boldsymbol{X}_{\mathsf{E}} \in \mathbb{F}_q^{n_{\mathsf{E}}\times L}$. Then for every k packets $\boldsymbol{y}_1,\ldots,\boldsymbol{y}_k$ that are linear combinations of the rows of $\boldsymbol{X}_{\mathsf{A}}$, if the subspace $\Pi_Y = \langle \boldsymbol{y}_1,\ldots,\boldsymbol{y}_k\rangle$ is orthogonal to $\langle \boldsymbol{X}_{\mathsf{E}}\rangle$ we have*

$$I(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_k;\boldsymbol{X}_{\mathsf{E}}) = 0. \tag{8.22}$$

Proof. The proof is stated in Appendix 8.B. □

8.4.1 Special Case: Achievability Scheme for Two Terminals

For simplicity and without loss of generality we assume that $n_{\mathsf{B}} \leq n_{\mathsf{A}}$ and $n_{\mathsf{E}} \leq n_{\mathsf{A}}$. The key generation scheme starts by Alice broadcasting a message $\boldsymbol{X}_{\mathsf{A}}[t]$ at time t of the form of (8.13). Then, Theorem 8.3 states that the secrecy rate $\mathfrak{R}_s^{\mathsf{non-coh}}$ is achievable if[4]

$$\mathfrak{R}_s^{\mathsf{non-coh}} \leq \big[\dim(U_{\mathsf{B}}+\Pi_{\mathsf{E}}) - \dim(\Pi_{\mathsf{E}})\big](L-n_{\mathsf{A}})\log q, \tag{8.23}$$

where $U_{\mathsf{B}} = \Pi_{\mathsf{B}} \setminus_s \Pi_{\mathsf{E}}$. Because $U_{\mathsf{B}} \cap \Pi_{\mathsf{E}} = \{\boldsymbol{0}\}$, we have

$$\begin{aligned}
\mathfrak{R}_s^{\mathsf{non-coh}} &\leq \big[\dim(U_{\mathsf{B}})\big](L-n_{\mathsf{A}})\log q \\
&= \big[\dim(\Pi_{\mathsf{B}}) - \dim(\Pi_{\mathsf{B}}\cap\Pi_{\mathsf{E}})\big](L-n_{\mathsf{A}})\log q \\
&= \big[n_{\mathsf{B}} - (n_{\mathsf{B}}+n_{\mathsf{E}}-n_{\mathsf{A}})^+\big](L-n_{\mathsf{A}})\log q \\
&= \big[\min[n_{\mathsf{A}}, n_{\mathsf{B}}+n_{\mathsf{E}}] - n_{\mathsf{E}}\big](L-n_{\mathsf{A}})\log q, \qquad \text{bits/channel use,}
\end{aligned} \tag{8.24}$$

where this is the same as the upper bound given in Theorem 8.2. This is obvious when $n_{\mathsf{A}} \leq n_{\mathsf{B}}+n_{\mathsf{E}}$. On the other hand, for the case where $n_{\mathsf{A}} > n_{\mathsf{B}}+n_{\mathsf{E}}$, we can reduce the number of injected packets by Alice in every time-slot from n_{A} to $n_{\mathsf{B}}+n_{\mathsf{E}}$ (there is no need to use more than $n_{\mathsf{B}}+n_{\mathsf{E}}$ degrees of freedom).

4. For the convenience of notation, we have replaced $U_{\{\mathsf{B}\}}$ with U_{B}.

Remark 8.2. *Note that in the above scheme, as long as $n_{\mathsf{E}} < n_{\mathsf{A}}$, the secrecy rate is non-zero.*

Now, we compare the derived secrecy rate with the case where no feedback is allowed. First let us assume that $n_{\mathsf{B}} \geq n_{\mathsf{E}}$. Then, in the non-coherent NC scenario introduced in Section 8.1, it can be easily verified that the channel from Alice to Eve is a stochastically degraded *(for the definition refer to [102, p. 373]) version of the channel from Alice to Bob.*

So by applying the result of [91] or [103, Theorem 3], for the secret key sharing capacity we can write

$$\begin{aligned} C_s &= \max_{P_{\boldsymbol{X}_{\mathsf{A}}}} \left[I(\boldsymbol{X}_{\mathsf{A}}; \boldsymbol{X}_{\mathsf{B}}) - I(\boldsymbol{X}_{\mathsf{A}}; \boldsymbol{X}_{\mathsf{E}}) \right] \\ &= \max_{P_{\Pi_{\mathsf{A}}}} \left[I(\Pi_{\mathsf{A}}; \Pi_{\mathsf{B}}) - I(\Pi_{\mathsf{A}}; \Pi_{\mathsf{E}}) \right], \end{aligned} \tag{8.25}$$

where the sufficiency of optimization over subspaces follows from Theorem 3.1. Similar to the proof of Lemma 8.4 (because in the proof of Lemma 8.4, we also maximize an expression that contains subtraction of two mutual information similar to (8.25)*), one can show that*

$$C_s = [n_{\mathsf{B}} - n_{\mathsf{E}}](L - n_{\mathsf{B}}) \log q, \tag{8.26}$$

which is positive only if $n_{\mathsf{B}} > n_{\mathsf{E}}$ (obviously for the case $n_{\mathsf{B}} < n_{\mathsf{E}}$ we have $C_s = 0$ as well, because even for a weaker eavesdropper, when $n_{\mathsf{B}} = n_{\mathsf{E}}$, we have $C_s = 0$).

The above comparison demonstrates the amount of improvement of the secret key generation rate we might gain by using feedback.

8.4.2 Special Case: Achievability Scheme for Three Terminals

As an another example, here we consider the three trusted terminals problem (i.e., $m = 2$). As before, we assume that $n_{\mathsf{A}} < L$ and for the convenience we consider the case where $n_{\mathsf{B}} = n_{\mathsf{C}} \leq n_{\mathsf{A}}$ and $n_{\mathsf{E}} \leq n_{\mathsf{A}}$.

In order to characterize the achievable secrecy rate, we need to find the dimension of subspaces U_{B}, U_{C}, and U_{BC} and their sums (including Π_{E}). We assume that the field size q is large. We know that Π_{B}, Π_{C}, and Π_{E} are chosen uniformly at random from n_{A}-dimensional space Π_{A}. Subspaces Π_{BC} and Π_{BE} are also distributed independently and uniformly at random in Π_{B}. Similarly, the same is true for Π_{BC} and Π_{CE} in Π_{C}.

From (8.14), we have

$$\begin{cases} U_{\mathsf{B}} = \Pi_{\mathsf{B}} \setminus_s (\Pi_{\mathsf{BC}} + \Pi_{\mathsf{BE}}) \\ U_{\mathsf{C}} = \Pi_{\mathsf{C}} \setminus_s (\Pi_{\mathsf{BC}} + \Pi_{\mathsf{CE}}) \\ U_{\mathsf{BC}} = \Pi_{\mathsf{BC}} \setminus_s (\Pi_{\mathsf{BCE}}), \end{cases} \tag{8.27}$$

so we can write

$$\begin{aligned}\dim(U_{\mathsf{B}}) &= \dim(\Pi_{\mathsf{B}}) - \dim(\Pi_{\mathsf{BC}} + \Pi_{\mathsf{BE}}) \\ &\overset{\text{(a)}}{=} \dim(\Pi_{\mathsf{B}}) - \min\left[\dim(\Pi_{\mathsf{BC}}) + \dim(\Pi_{\mathsf{BE}}), \dim(\Pi_{\mathsf{B}})\right] \\ &\overset{\text{(b)}}{=} n_{\mathsf{B}} - \min\left[\dim(\Pi_{\mathsf{BC}}) + \dim(\Pi_{\mathsf{BE}}), n_{\mathsf{B}}\right] \\ &= \left[n_{\mathsf{B}} - \dim(\Pi_{\mathsf{BC}}) - \dim(\Pi_{\mathsf{BE}})\right]^+ \\ &\overset{\text{(c)}}{=} \left[n_{\mathsf{B}} - (2n_{\mathsf{B}} - n_{\mathsf{A}})^+ - (n_{\mathsf{B}} + n_{\mathsf{E}} - n_{\mathsf{A}})^+\right]^+, \end{aligned} \tag{8.28}$$

where (a) follows from Lemma 2.11 because Π_{BC} and Π_{BE} are chosen independently and uniformly at random from Π_{B}, (b) is true because q is large, and (c) follows from Lemma 2.11. Note that because we have assumed $n_{\mathsf{B}} = n_{\mathsf{C}}$ it follows that $\dim(U_{\mathsf{C}}) = \dim(U_{\mathsf{B}})$.

Similarly, for the dimension of U_{BC} we can write

$$\begin{aligned}\dim(U_{\mathsf{BC}}) &= \dim(\Pi_{\mathsf{BC}}) - \dim(\Pi_{\mathsf{BCE}}) \\ &= \dim(\Pi_{\mathsf{BC}}) - \left[\dim(\Pi_{\mathsf{BC}}) + n_{\mathsf{E}} - n_{\mathsf{A}}\right]^+ \\ &= \min\left[n_{\mathsf{A}} - n_{\mathsf{E}}, \dim(\Pi_{\mathsf{BC}})\right] \\ &= \min\left[n_{\mathsf{A}} - n_{\mathsf{E}}, (2n_{\mathsf{B}} - n_{\mathsf{A}})^+\right]. \end{aligned} \tag{8.29}$$

Proposition 8.1. *From the construction, the subspaces U_{B}, U_{C}, and U_{BC} are orthogonal. The same holds for U_{B}, U_{BC}, and Π_{E} and similarly for U_{C}, U_{BC}, and Π_{E} w.h.p.*

Now we may write the optimization problem stated in Theorem 8.3 as follows

$$\begin{array}{ll} \text{maximize} & \min\left[\theta_{\mathsf{B}} + \theta_{\mathsf{BC}}, \theta_{\mathsf{C}} + \theta_{\mathsf{BC}}\right](L - n_{\mathsf{A}}) \log q \\ \text{subject to} & \theta_{\mathsf{B}} \le \dim(U_{\mathsf{B}} + \Pi_{\mathsf{E}}) - n_{\mathsf{E}} \\ & \theta_{\mathsf{C}} \le \dim(U_{\mathsf{C}} + \Pi_{\mathsf{E}}) - n_{\mathsf{E}} \\ & \theta_{\mathsf{BC}} \le \dim(U_{\mathsf{BC}} + \Pi_{\mathsf{E}}) - n_{\mathsf{E}} \\ & \theta_{\mathsf{B}} + \theta_{\mathsf{C}} \le \dim(U_{\mathsf{B}} + U_{\mathsf{C}} + \Pi_{\mathsf{E}}) - n_{\mathsf{E}} \\ & \theta_{\mathsf{B}} + \theta_{\mathsf{C}} + \theta_{\mathsf{BC}} \le \dim(U_{\mathsf{B}} + U_{\mathsf{C}} + U_{\mathsf{BC}} + \Pi_{\mathsf{E}}) - n_{\mathsf{E}}. \end{array} \tag{8.30}$$

Because of the symmetry in the problem ($n_{\mathsf{B}} = n_{\mathsf{C}}$), for the optimal solution we should have $\theta_{\mathsf{B}} = \theta_{\mathsf{C}}$. Knowing this and using Proposition 8.1, we may further simplify the above linear program as follows

$$\begin{array}{ll} \text{maximize} & [\theta_{\mathsf{B}} + \theta_{\mathsf{BC}}]\,(L - n_{\mathsf{A}}) \log q \\ \text{subject to} & \theta_{\mathsf{B}} \le \frac{1}{2}\left[\dim(U_{\mathsf{B}} + U_{\mathsf{C}} + \Pi_{\mathsf{E}}) - n_{\mathsf{E}}\right] \triangleq \alpha_1 \\ & \theta_{\mathsf{BC}} \le \dim(U_{\mathsf{BC}}) \triangleq \alpha_2 \\ & 2\theta_{\mathsf{B}} + \theta_{\mathsf{BC}} \le \dim(U_{\mathsf{B}} + U_{\mathsf{C}} + U_{\mathsf{BC}} + \Pi_{\mathsf{E}}) - n_{\mathsf{E}} \triangleq \alpha_3. \end{array} \tag{8.31}$$

From the definitions of α's, we can easily observe that, $\alpha_3 \ge 2\alpha_1$, $\alpha_3 \ge \alpha_2$, and $\alpha_3 \le 2\alpha_1 + \alpha_2$. Hence, $\theta_{\mathsf{B}} + \theta_{\mathsf{BC}}$ gets its maximum at the point $(\theta_{\mathsf{B}}, \theta_{\mathsf{BC}}) =$

$(\frac{\alpha_3-\alpha_2}{2}, \alpha_2)$. Thus, for the maximum achievable secrecy rate we have

$$\mathfrak{R}_s^{\mathsf{non-coh}} = \left[\frac{\alpha_2+\alpha_3}{2}\right](L-n_{\mathsf{A}})\log q. \tag{8.32}$$

As mentioned before, we assume that subspaces $U_{\mathcal{J}}$'s are chosen uniformly at random from $\Pi_{\mathcal{J}}$. So Π_{E} and $U_{\mathcal{J}}$'s are independent and for α_3, w.h.p., we can write

$$\begin{aligned}\alpha_3 &= \min\big[\dim(U_{\mathsf{B}})+\dim(U_{\mathsf{C}})+\dim(U_{\mathsf{BC}})+\dim(\Pi_{\mathsf{E}}), n_{\mathsf{A}}\big]-n_{\mathsf{E}}\\ &= \min\big[\dim(U_{\mathsf{B}})+\dim(U_{\mathsf{C}})+\dim(U_{\mathsf{BC}}), n_{\mathsf{A}}-n_{\mathsf{E}}\big]\\ &= \min\big[2\dim(U_{\mathsf{B}})+\dim(U_{\mathsf{BC}}), n_{\mathsf{A}}-n_{\mathsf{E}}\big].\end{aligned} \tag{8.33}$$

Finally, for the secrecy rate (achievable asymptotically when q goes to infinity) we have

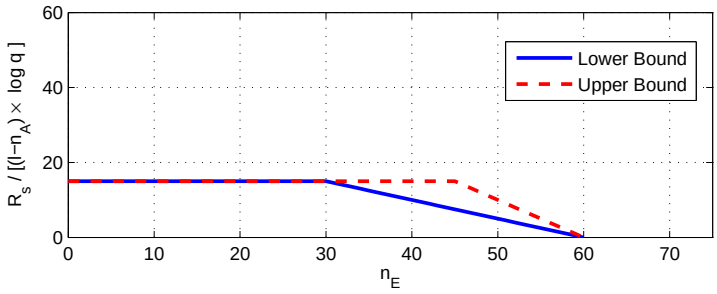
$$\mathfrak{R}_s^{\mathsf{non-coh}} = \min\left[\dim(U_{\mathsf{B}})+\dim(U_{\mathsf{BC}}), \frac{1}{2}\left(n_{\mathsf{A}}+\dim(U_{\mathsf{BC}})-n_{\mathsf{E}}\right)\right](L-n_{\mathsf{A}})\log q. \tag{8.34}$$

Example 8.1. *As an example, here we compare the achievable secret key sharing rate among three legitimate terminals (i.e., $m=2$) as derived in* (8.34) *with the upper bound stated in Theorem 8.2. We consider two symmetric setup where for the first one we have $n_{\mathsf{A}}=60$, $n_{\mathsf{B}}=n_{\mathsf{C}}=15$ (see Figure 8.2a) and for the second one we have $n_{\mathsf{A}}=60$, $n_{\mathsf{B}}=n_{\mathsf{C}}=45$ (see Figure 8.2b). In each of these situations, we depict the upper and lower bounds on the secret key generation rate as a function of the number of packets (degrees of freedom) received by Eve, i.e., n_{E}.*

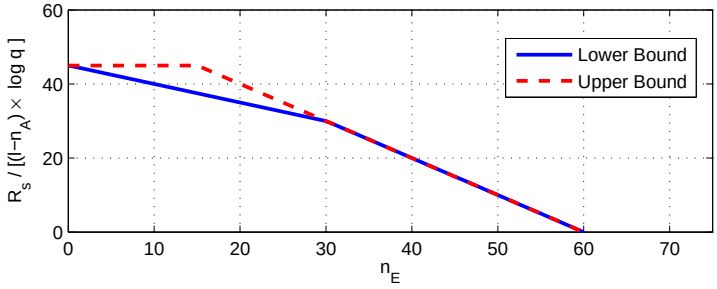
8.5 Concluding Remarks

In this chapter, we have considered the problem of sharing a secret key among multiple terminals in the presence of a passive eavesdropper. The trusted nodes have access to a non-coherent multicast NC channel and also are able to discuss over a public channel which is overheard by every nodes.

For this setup, we have derived upper and lower bounds for the secrecy capacity. The proposed achievability scheme is based on subspace coding and it works for arbitrary number of terminals. However, it should be mentioned that this result holds asymptotically in the field size.



(a) $m = 2$, $n_{\mathsf{A}} = 60$, and $n_{\mathsf{B}} = n_{\mathsf{C}} = 15$.



(b) $m = 2$, $n_{\mathsf{A}} = 60$, and $n_{\mathsf{B}} = n_{\mathsf{C}} = 45$.

Figure 8.2 – A comparison between the achievable secrecy rate of Theorem 8.3 and the upper bound given by Theorem 8.2 for two cases: (a) when $m = 2$, $n_{\mathsf{A}} = 60$, and $n_{\mathsf{B}} = n_{\mathsf{C}} = 15$ and (b) when $m = 2$, $n_{\mathsf{A}} = 60$, and $n_{\mathsf{B}} = n_{\mathsf{C}} = 45$.

8.A Simplifying the Mutual Information for a Non-coherent NC Channel

Let us consider a non-coherent NC channel described by the following matrix channel

$$\boldsymbol{Y}[t] = \boldsymbol{F}[t]\boldsymbol{X}[t], \tag{8.35}$$

where $\boldsymbol{X}[t] \in \mathbb{F}_q^{n_x \times L}$, $\boldsymbol{Y}[t] \in \mathbb{F}_q^{n_y \times L}$, and $\boldsymbol{F}[t] \in \mathbb{F}_q^{n_y \times n_x}$ is an uniformly at random chosen transfer matrix which is independently chosen for every time-slot t. For simplicity we assume that $L \geq \max[n_x, n_y]$.

As stated in Theorem 3.1, in order to find the capacity of (8.35) we can instead focus on an equivalent subspace channel described by a transition probability as follows

$$P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x) \triangleq \begin{cases} \psi(n_y, \dim(\pi_y)) q^{-n_y \dim(\pi_x)} & \text{if } \pi_y \sqsubseteq \pi_x, \\ 0 & \text{otherwise.} \end{cases} \tag{8.36}$$

In this work we focus on large q regime, so we can approximate the above

transition probability as follows

$$P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x) = \mathbb{1}_{\{\dim \pi_x \le n_y\}} \mathbb{1}_{\{\pi_y = \pi_x\}} + \frac{\mathbb{1}_{\{\dim \pi_x > n_y\}} \mathbb{1}_{\{\dim \pi_y = n_y\}} \mathbb{1}_{\{\pi_y \sqsubseteq \pi_x\}}}{\begin{bmatrix} \dim \pi_x \\ n_y \end{bmatrix}}. \tag{8.37}$$

From here on we assume that the input distribution is uniform over all subspaces having the same dimension, namely

$$\mathbb{P}\left[\Pi_X = \pi_x\right] = \alpha_{d_x} \begin{bmatrix} L \\ d_x \end{bmatrix}^{-1}, \tag{8.38}$$

where $d_x = \dim \pi_x$ and $\alpha_{d_x} = \mathbb{P}\left[\dim \Pi_X = d_x\right]$.

Then, for P_{Π_Y} we can write

$$\begin{aligned} P_{\Pi_Y}(\pi_y) &= \sum_{\pi_x} P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x) P_{\Pi_X}(\pi_x) \\ &= \sum_{\substack{\pi_x: \\ \dim \pi_x \le \min[n_x, n_y]}} \mathbb{1}_{\{\pi_y = \pi_x\}} P_{\Pi_X}(\pi_x) \\ &\quad + \sum_{\substack{\pi_x: \\ n_y < \dim \pi_x \le n_x}} \frac{\mathbb{1}_{\{\dim \pi_y = n_y\}} \mathbb{1}_{\{\pi_y \sqsubseteq \pi_x\}}}{\begin{bmatrix} \dim \pi_x \\ n_y \end{bmatrix}} P_{\Pi_X}(\pi_x). \end{aligned} \tag{8.39}$$

So we have

$$\begin{aligned} P_{\Pi_Y}(\pi_y) &= P_{\Pi_X}(\pi_y) \mathbb{1}_{\{\dim \pi_y \le \min[n_x, n_y]\}} + \sum_{d_x = n_y + 1}^{n_x} \sum_{\substack{\pi_x: \\ \pi_y \sqsubseteq \pi_x, \\ \dim \pi_x = d_x}} \frac{\mathbb{1}_{\{\dim \pi_y = n_y\}}}{\begin{bmatrix} d_x \\ n_y \end{bmatrix}} \cdot \frac{\alpha_{d_x}}{\begin{bmatrix} L \\ d_x \end{bmatrix}} \\ &= P_{\Pi_X}(\pi_y) \mathbb{1}_{\{\dim \pi_y \le \min[n_x, n_y]\}} + \sum_{d_x = n_y + 1}^{n_x} \begin{bmatrix} L - n_y \\ d_x - n_y \end{bmatrix} \frac{\mathbb{1}_{\{\dim \pi_y = n_y\}}}{\begin{bmatrix} d_x \\ n_y \end{bmatrix}} \cdot \frac{\alpha_{d_x}}{\begin{bmatrix} L \\ d_x \end{bmatrix}}. \end{aligned} \tag{8.40}$$

Now, we use the following relation (see Lemma 2.2) to further simplify the expression in front of the summation

$$\begin{bmatrix} L - n_y \\ d_x - n_y \end{bmatrix} \begin{bmatrix} L \\ n_y \end{bmatrix} = \begin{bmatrix} d_x \\ n_y \end{bmatrix} \begin{bmatrix} L \\ d_x \end{bmatrix}. \tag{8.41}$$

Then for P_{Π_Y} we have

$$\begin{aligned} & P_{\Pi_Y}(\pi_y) = \\ &= P_{\Pi_X}(\pi_y)\mathbb{1}_{\{\dim \pi_y \leq \min[n_x, n_y]\}} + \sum_{d_x=n_y+1}^{n_x} \mathbb{1}_{\{\dim \pi_y = n\}} \alpha_{d_x} \begin{bmatrix} L \\ n_y \end{bmatrix}^{-1} \\ &= P_{\Pi_X}(\pi_y)\mathbb{1}_{\{\dim \pi_y \leq \min[n_x, n_y]\}} + \mathbb{P}\left[\dim \Pi_X > n_y\right] \mathbb{1}_{\{\dim \pi_y = n_y\}} \begin{bmatrix} L \\ n_y \end{bmatrix}^{-1} \\ &= P_{\Pi_X}(\pi_y)\mathbb{1}_{\{\dim \pi_y \leq \min[n_x, n_y - 1]\}} + \mathbb{P}\left[\dim \Pi_X \geq n_y\right] \mathbb{1}_{\{\dim \pi_y = n_y\}} \begin{bmatrix} L \\ n_y \end{bmatrix}^{-1}. \end{aligned} \tag{8.42}$$

Hence, by definition, for the mutual information $I(\Pi_X; \Pi_Y)$ we can write

$$I(\Pi_X; \Pi_Y) = \sum_{d_x=0}^{n_x} \sum_{\substack{\pi_x: \\ \dim \pi_x = d_x}} \sum_{d_y=0}^{\min[n_y, d_x]} \sum_{\substack{\pi_y: \\ \dim \pi_y = d_y, \\ \pi_y \sqsubseteq \pi_x}} F(\pi_x, \pi_y) \tag{8.43}$$

where

$$F(\pi_x, \pi_y) = P_{\Pi_X}(\pi_x) P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x) \log \frac{P_{\Pi_Y|\Pi_X}(\pi_y|\pi_x)}{P_{\Pi_Y}(\pi_y)}. \tag{8.44}$$

Then we have

$$\begin{aligned} I(\Pi_X; \Pi_Y) = & \sum_{d_x=0}^{\min[n_x, n_y]} \sum_{\substack{\pi_x: \\ \dim \pi_x = d_x}} \sum_{d_y=0}^{d_x} \sum_{\substack{\pi_y: \\ \dim \pi_y = d_y, \\ \pi_y \sqsubseteq \pi_x}} \frac{\alpha_{d_x}}{\begin{bmatrix} L \\ d_x \end{bmatrix}} \mathbb{1}_{\{\pi_y = \pi_x\}} \log \frac{\mathbb{1}_{\{\pi_y = \pi_x\}}}{P_{\Pi_Y}(\pi_y)}, \\ & + \mathbb{1}_{\{n_y < n_x\}} \left\{ \sum_{d_x=n_y+1}^{n_x} \sum_{\substack{\pi_x: \\ \dim \pi_x = d_x}} \sum_{d_y=0}^{n_y} \sum_{\substack{\pi_y: \\ \dim \pi_y = d_y, \\ \pi_y \sqsubseteq \pi_x}} F'(d_x, d_y, \pi_y) \right\} \end{aligned} \tag{8.45}$$

where

$$F'(d_x, d_y, \pi_y) = \frac{\alpha_{d_x}}{\begin{bmatrix} L \\ d_x \end{bmatrix}} \frac{\mathbb{1}_{\{d_x > n_y\}} \mathbb{1}_{\{d_y = n_y\}}}{\begin{bmatrix} d_x \\ n_y \end{bmatrix}} \log \frac{\mathbb{1}_{\{d_x > n_y\}} \mathbb{1}_{\{d_y = n_y\}}}{P_{\Pi_Y}(\pi_y) \begin{bmatrix} d_x \\ n_y \end{bmatrix}}. \tag{8.46}$$

This simplifies to

$$I(\Pi_X;\Pi_Y) = -\sum_{d_x=0}^{\min[n_x,n_y]} \sum_{\substack{\pi_x:\\ \dim\pi_x=d_x}} \frac{\alpha_{d_x}}{\begin{bmatrix} L \\ d_x \end{bmatrix}} \log P_{\Pi_Y}(\pi_x) - \mathbb{1}_{\{n_y<n_x\}} \left\{ \sum_{d_x=n_y+1}^{n_x} \sum_{\substack{\pi_x:\\ \dim\pi_x=d_x}} \frac{\alpha_{d_x}}{\begin{bmatrix} L \\ d_x \end{bmatrix}} \log \left[P_{\Pi_Y}(\pi_y) \begin{bmatrix} d_x \\ n_y \end{bmatrix} \right] \right\}, \tag{8.47}$$

where in the last line π_y is an arbitrary subspace with $\dim \pi_y = n_y$. So finally we can write

$$I(\Pi_X;\Pi_Y) = -\sum_{d_x=0}^{\min[n_x,n_y]} \alpha_{d_x} \log \frac{\alpha_{d_x}}{\begin{bmatrix} L \\ d_x \end{bmatrix}} - \mathbb{1}_{\{n_y<n_x\}} \left\{ \sum_{d_x=n_y+1}^{n_x} \alpha_{d_x} \log \left[\frac{\begin{bmatrix} d_x \\ n_y \end{bmatrix}}{\begin{bmatrix} L \\ n_y \end{bmatrix}} \mathbb{P}\left[\dim \Pi_X \geq n_y\right] \right] \right\}. \tag{8.48}$$

8.B Omitted Proofs

Proof of Lemma 8.1. Let us expand $I(\boldsymbol{X}_\mathsf{A};\boldsymbol{X}_\mathsf{B}|\boldsymbol{X}_\mathsf{E})$ as follows [5]

$$\begin{aligned} I(\boldsymbol{X}_\mathsf{A};\boldsymbol{X}_\mathsf{B}|\boldsymbol{X}_\mathsf{E}) &= \sum_{\boldsymbol{x}_\mathsf{A},\boldsymbol{x}_\mathsf{B},\boldsymbol{x}_\mathsf{E}} p(\boldsymbol{x}_\mathsf{A},\boldsymbol{x}_\mathsf{B},\boldsymbol{x}_\mathsf{E}) \log \frac{p(\boldsymbol{x}_\mathsf{A},\boldsymbol{x}_\mathsf{B}|\boldsymbol{x}_\mathsf{E})}{p(\boldsymbol{x}_\mathsf{A}|\boldsymbol{x}_\mathsf{E})p(\boldsymbol{x}_\mathsf{B}|\boldsymbol{x}_\mathsf{E})} \\ &= \sum_{\boldsymbol{x}_\mathsf{A},\boldsymbol{x}_\mathsf{B},\boldsymbol{x}_\mathsf{E}} p(\boldsymbol{x}_\mathsf{A})p(\boldsymbol{x}_\mathsf{B},\boldsymbol{x}_\mathsf{E}|\boldsymbol{x}_\mathsf{A}) \log \frac{p(\boldsymbol{x}_\mathsf{B},\boldsymbol{x}_\mathsf{E}|\boldsymbol{x}_\mathsf{A})}{p(\boldsymbol{x}_\mathsf{E}|\boldsymbol{x}_\mathsf{A})p(\boldsymbol{x}_\mathsf{B}|\boldsymbol{x}_\mathsf{E})}. \end{aligned} \tag{8.49}$$

From here we will use the independence of the channels from Alice to Bob and to Eve, $p(\boldsymbol{x}_\mathsf{B},\boldsymbol{x}_\mathsf{E}|\boldsymbol{x}_\mathsf{A}) = p(\boldsymbol{x}_\mathsf{B}|\boldsymbol{x}_\mathsf{A})p(\boldsymbol{x}_\mathsf{E}|\boldsymbol{x}_\mathsf{A})$, so we have

$$\begin{aligned} I(\boldsymbol{X}_\mathsf{A};\boldsymbol{X}_\mathsf{B}|\boldsymbol{X}_\mathsf{E}) &= \sum_{\boldsymbol{x}_\mathsf{A},\boldsymbol{x}_\mathsf{B},\boldsymbol{x}_\mathsf{E}} p(\boldsymbol{x}_\mathsf{A})p(\boldsymbol{x}_\mathsf{B},\boldsymbol{x}_\mathsf{E}|\boldsymbol{x}_\mathsf{A}) \log \frac{p(\boldsymbol{x}_\mathsf{B}|\boldsymbol{x}_\mathsf{A})}{p(\boldsymbol{x}_\mathsf{B}|\boldsymbol{x}_\mathsf{E})} \\ &= \sum_{\boldsymbol{x}_\mathsf{A},\boldsymbol{x}_\mathsf{B},\boldsymbol{x}_\mathsf{E}} p(\boldsymbol{x}_\mathsf{A})p(\boldsymbol{x}_\mathsf{B},\boldsymbol{x}_\mathsf{E}|\boldsymbol{x}_\mathsf{A}) \log \left[p(\boldsymbol{x}_\mathsf{B}|\boldsymbol{x}_\mathsf{A}) \frac{p(\boldsymbol{x}_\mathsf{E})}{p(\boldsymbol{x}_\mathsf{B},\boldsymbol{x}_\mathsf{E})} \right] \\ &= -H(\boldsymbol{X}_\mathsf{B}|\boldsymbol{X}_\mathsf{A}) + \sum_{\boldsymbol{x}_\mathsf{A},\boldsymbol{x}_\mathsf{B},\boldsymbol{x}_\mathsf{E}} p(\boldsymbol{x}_\mathsf{A})p(\boldsymbol{x}_\mathsf{B},\boldsymbol{x}_\mathsf{E}|\boldsymbol{x}_\mathsf{A}) \log \frac{p(\boldsymbol{x}_\mathsf{E})}{p(\boldsymbol{x}_\mathsf{B},\boldsymbol{x}_\mathsf{E})} \\ &= -H(\boldsymbol{X}_\mathsf{B}|\boldsymbol{X}_\mathsf{A}) - \sum_{\boldsymbol{x}_\mathsf{B},\boldsymbol{x}_\mathsf{E}} f\big(p(\boldsymbol{x}_\mathsf{A}), p(\boldsymbol{x}_\mathsf{B}|\boldsymbol{x}_\mathsf{A}), p(\boldsymbol{x}_\mathsf{E}|\boldsymbol{x}_\mathsf{A})\big), \end{aligned} \tag{8.50}$$

5. Note that without loss of generality, we have replaced $\boldsymbol{X}_i$ with $\boldsymbol{X}_\mathsf{B}$.

where

$$f\big(p(\boldsymbol{x}_\mathsf{A}), p(\boldsymbol{x}_\mathsf{B}|\boldsymbol{x}_\mathsf{A}), p(\boldsymbol{x}_\mathsf{E}|\boldsymbol{x}_\mathsf{A})\big) \triangleq \sum_{\boldsymbol{x}_\mathsf{A}} p(\boldsymbol{x}_\mathsf{A}) p(\boldsymbol{x}_\mathsf{B}, \boldsymbol{x}_\mathsf{E}|\boldsymbol{x}_\mathsf{A}) \log \frac{\sum_{\boldsymbol{x}''_\mathsf{A}} p(\boldsymbol{x}_\mathsf{B}, \boldsymbol{x}_\mathsf{E}|\boldsymbol{x}''_\mathsf{A}) p(\boldsymbol{x}''_\mathsf{A})}{\sum_{\boldsymbol{x}'_\mathsf{A}} p(\boldsymbol{x}_\mathsf{E}|\boldsymbol{x}'_\mathsf{A}) p(\boldsymbol{x}'_\mathsf{A})}. \quad (8.51)$$

Suppose $p_1(\boldsymbol{x}_\mathsf{A})$ and $p_2(\boldsymbol{x}_\mathsf{A})$ are two arbitrary distributions over random variable $\boldsymbol{X}_\mathsf{A}$. Let us define $p_\lambda(\boldsymbol{x}_\mathsf{A}) \triangleq \lambda p_1(\boldsymbol{x}_\mathsf{A}) + (1-\lambda) p_2(\boldsymbol{x}_\mathsf{A})$ where $0 \leq \lambda \leq 1$. Using the log-sum inequality (see [38], Theorem 2.7.1), we can write

$$f\big(p_\lambda(\boldsymbol{x}_\mathsf{A}), p(\boldsymbol{x}_\mathsf{B}, \boldsymbol{x}_\mathsf{E}|\boldsymbol{x}_\mathsf{A})\big) \leq \lambda f\big(p_1(\boldsymbol{x}_\mathsf{A}), p(\boldsymbol{x}_\mathsf{B}, \boldsymbol{x}_\mathsf{E}|\boldsymbol{x}_\mathsf{A})\big) + (1-\lambda) f\big(p_2(\boldsymbol{x}_\mathsf{A}), p(\boldsymbol{x}_\mathsf{B}, \boldsymbol{x}_\mathsf{E}|\boldsymbol{x}_\mathsf{A})\big), \quad (8.52)$$

which shows that f is a convex function in $p(\boldsymbol{x}_\mathsf{A})$. we also know that $H(\boldsymbol{X}_\mathsf{B}|\boldsymbol{X}_\mathsf{A})$ is a linear function with respect to $p(\boldsymbol{x}_\mathsf{A})$ so $I(\boldsymbol{X}_\mathsf{A}; \boldsymbol{X}_\mathsf{B}|\boldsymbol{X}_\mathsf{E})$ is a concave function with respect to $p(\boldsymbol{x}_\mathsf{A})$. □

Proof of Lemma 8.4. By using Lemma 8.1 and Lemma 8.2, we conclude that in order to maximize $I(\boldsymbol{X}_\mathsf{A}; \boldsymbol{X}_i|\boldsymbol{X}_\mathsf{E})$ with respect to $P_{\boldsymbol{X}_\mathsf{A}}$, it is sufficient to maximize $I(\Pi_\mathsf{A}; \Pi_i|\Pi_\mathsf{E})$ for and equivalent subspace channel introduced in (8.8). Also Lemma 8.3 indicates that considering input distributions that are uniform over all subspaces having the same dimension is sufficient.

Let us assume

$$\mathbb{P}\left[\Pi_\mathsf{A} = \pi_\mathsf{A}\right] = \alpha_d \begin{bmatrix} L \\ d \end{bmatrix}^{-1}, \quad (8.53)$$

where $d = \dim \pi_\mathsf{A}$ and $\alpha_d = \mathbb{P}\left[\dim \Pi_\mathsf{A} = d\right]$. Now, define

$$f \triangleq I(\Pi_\mathsf{A}; \Pi_i|\Pi_\mathsf{E}) = I(\Pi_\mathsf{A}; \Pi_i, \Pi_\mathsf{E}) - I(\Pi_\mathsf{A}; \Pi_\mathsf{E}), \quad (8.54)$$

and the goal is to maximize f with respect to α_is.

We consider two cases as follows.

First case: $n_i + n_\mathsf{E} \leq n_\mathsf{A}$

Then, by applying the results of Appendix 8.A, specially (8.48), we can write

$$f = -\sum_{d=n_\mathsf{E}+1}^{n_i+n_\mathsf{E}} \alpha_d \log \frac{\alpha_d}{\begin{bmatrix} L \\ d \end{bmatrix}} - \sum_{d=n_i+n_\mathsf{E}+1}^{n_\mathsf{A}} \alpha_d \log \left[\frac{\begin{bmatrix} d \\ n_i+n_\mathsf{E} \end{bmatrix}}{\begin{bmatrix} L \\ n_i+n_\mathsf{E} \end{bmatrix}} \mathbb{P}\left[\dim \Pi_\mathsf{A} \geq n_i + n_\mathsf{E}\right] \right] + \sum_{d=n_\mathsf{E}+1}^{n_\mathsf{A}} \alpha_d \log \left[\frac{\begin{bmatrix} d \\ n_\mathsf{E} \end{bmatrix}}{\begin{bmatrix} L \\ n_\mathsf{E} \end{bmatrix}} \mathbb{P}\left[\dim \Pi_\mathsf{A} \geq n_\mathsf{E}\right] \right]. \quad (8.55)$$

Now we have to maximize f with respect to the input distribution, α_i. We know that the mutual information is a concave function with respect to α_i's. This allows us to use the Kuhn-Tucker theorem to solve the convex optimization problem. According to this theorem, the set of probabilities α_i^*, $0 \leq i \leq n_\mathsf{A}$,

maximize the mutual information if and only if there exists some constant λ such that

$$\begin{cases} \left.\frac{\partial f}{\partial \alpha_k}\right|_{\boldsymbol{\alpha}^*} = \lambda \quad \forall k: \ \alpha_k^* > 0, \\ \left.\frac{\partial f}{\partial \alpha_k}\right|_{\boldsymbol{\alpha}^*} \leq \lambda \quad \forall k: \ \alpha_k^* = 0, \end{cases} \tag{8.56}$$

where $0 \leq k \leq n_{\mathsf{A}}$, $\sum_{i=0}^{n_{\mathsf{A}}} \alpha_i^* = 1$, and $\boldsymbol{\alpha}^*$ is the vector of the optimum input probabilities of choosing subspaces of certain dimension,

$$\boldsymbol{\alpha}^* = \begin{bmatrix} \alpha_0^* & \cdots & \alpha_{n_{\mathsf{A}}}^* \end{bmatrix}^{\mathrm{T}}. \tag{8.57}$$

Taking the derivative for $0 \leq k < n_{\mathsf{E}}$ we have

$$\frac{\partial f}{\partial \alpha_k} = 0, \tag{8.58}$$

for $k = n_{\mathsf{E}}$ we have

$$\frac{\partial f}{\partial \alpha_k} = \sum_{d=n_{\mathsf{E}}+1}^{n_{\mathsf{A}}} \alpha_d \left[\frac{\log e}{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} \geq n_{\mathsf{E}}\right]} \right] = \frac{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} > n_{\mathsf{E}}\right]}{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} \geq n_{\mathsf{E}}\right]} \log e, \tag{8.59}$$

for $n_{\mathsf{E}} < k < n_i + n_{\mathsf{E}}$ we have

$$\frac{\partial f}{\partial \alpha_k} = -\log \frac{\alpha_k}{\begin{bmatrix} L \\ k \end{bmatrix}} - \log e + \log \left[\frac{\begin{bmatrix} k \\ n_{\mathsf{E}} \end{bmatrix}}{\begin{bmatrix} L \\ n_{\mathsf{E}} \end{bmatrix}} \mathbb{P}\left[\dim \Pi_{\mathsf{A}} \geq n_{\mathsf{E}}\right] \right] + \frac{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} > n_{\mathsf{E}}\right]}{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} \geq n_{\mathsf{E}}\right]} \log e, \tag{8.60}$$

for $k = n_i + n_{\mathsf{E}}$ we have

$$\begin{aligned} \frac{\partial f}{\partial \alpha_k} = &-\log \frac{\alpha_k}{\begin{bmatrix} L \\ k \end{bmatrix}} - \log e + \log \left[\frac{\begin{bmatrix} k \\ n_{\mathsf{E}} \end{bmatrix}}{\begin{bmatrix} L \\ n_{\mathsf{E}} \end{bmatrix}} \mathbb{P}\left[\dim \Pi_{\mathsf{A}} \geq n_{\mathsf{E}}\right] \right] + \frac{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} > n_{\mathsf{E}}\right]}{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} \geq n_{\mathsf{E}}\right]} \log e \\ &- \frac{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} > n_i + n_{\mathsf{E}}\right]}{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} \geq n_i + n_{\mathsf{E}}\right]} \log e, \end{aligned} \tag{8.61}$$

and finally for $n_i + n_{\mathsf{E}} < k \leq n_{\mathsf{A}}$ we have

$$\begin{aligned} \frac{\partial f}{\partial \alpha_k} = &+\log \left[\frac{\begin{bmatrix} k \\ n_{\mathsf{E}} \end{bmatrix}}{\begin{bmatrix} L \\ n_{\mathsf{E}} \end{bmatrix}} \mathbb{P}\left[\dim \Pi_{\mathsf{A}} \geq n_{\mathsf{E}}\right] \right] + \frac{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} > n_{\mathsf{E}}\right]}{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} \geq n_{\mathsf{E}}\right]} \log e \\ &- \log \left[\frac{\begin{bmatrix} k \\ n_i + n_{\mathsf{E}} \end{bmatrix}}{\begin{bmatrix} L \\ n_i + n_{\mathsf{E}} \end{bmatrix}} \mathbb{P}\left[\dim \Pi_{\mathsf{A}} \geq n_i + n_{\mathsf{E}}\right] \right] - \frac{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} > n_i + n_{\mathsf{E}}\right]}{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} \geq n_i + n_{\mathsf{E}}\right]} \log e. \end{aligned} \tag{8.62}$$

We can easily check that for large q, the input distribution that has $\alpha_{n_i+n_{\mathsf{E}}} = 1$ and $\alpha_i = 0$ for $i \neq n_i + n_{\mathsf{E}}$ satisfies the Kuhn-Tucker conditions. For this

distribution, we have

$$\begin{cases} 0 \le k < n_{\mathsf{E}} & : \ \frac{\partial f}{\partial \alpha_k} = 0 < \lambda, \\ k = n_{\mathsf{E}} & : \ \frac{\partial f}{\partial \alpha_k} = \log e < \lambda, \\ n_{\mathsf{E}} < k < n_i + n_{\mathsf{E}} & : \ \frac{\partial f}{\partial \alpha_k} = \log \begin{bmatrix} k \\ n_{\mathsf{E}} \end{bmatrix} \begin{bmatrix} L \\ n_{\mathsf{E}} \end{bmatrix}^{-1} < \lambda, \\ k = n_i + n_{\mathsf{E}} & : \ \frac{\partial f}{\partial \alpha_k} = \log \begin{bmatrix} L \\ n_i + n_{\mathsf{E}} \end{bmatrix} \begin{bmatrix} n_i + n_{\mathsf{E}} \\ n_{\mathsf{E}} \end{bmatrix} \begin{bmatrix} L \\ n_{\mathsf{E}} \end{bmatrix}^{-1} = \lambda, \\ n_i + n_{\mathsf{E}} < k \le n_{\mathsf{A}} & : \ \frac{\partial f}{\partial \alpha_k} = \log e + \log \begin{bmatrix} k \\ n_{\mathsf{E}} \end{bmatrix} \begin{bmatrix} L \\ n_{\mathsf{E}} \end{bmatrix}^{-1} \\ & \qquad\qquad - \log \begin{bmatrix} k \\ n_i + n_{\mathsf{E}} \end{bmatrix} \begin{bmatrix} L \\ n_i + n_{\mathsf{E}} \end{bmatrix}^{-1} < \lambda. \end{cases} \tag{8.63}$$

So we have

$$\begin{aligned} \max_{P_{\Pi_{\mathsf{A}}}} I(\Pi_{\mathsf{A}}; \Pi_i | \Pi_{\mathsf{E}}) &= \lambda \\ &= \log \begin{bmatrix} L \\ n_i + n_{\mathsf{E}} \end{bmatrix} \begin{bmatrix} n_i + n_{\mathsf{E}} \\ n_{\mathsf{E}} \end{bmatrix} \begin{bmatrix} L \\ n_{\mathsf{E}} \end{bmatrix}^{-1} \\ &= \log \begin{bmatrix} L - n_{\mathsf{E}} \\ n_i \end{bmatrix} \\ &\approx n_i (L - n_i - n_{\mathsf{E}}) \log q. \end{aligned} \tag{8.64}$$

Second case: $n_i + n_{\mathsf{E}} > n_{\mathsf{A}}$
For this case the function f becomes

$$f = -\sum_{d=n_{\mathsf{E}}+1}^{n_{\mathsf{A}}} \alpha_d \log \frac{\alpha_d}{\begin{bmatrix} L \\ d \end{bmatrix}} + \sum_{d=n_{\mathsf{E}}+1}^{n_{\mathsf{A}}} \alpha_d \log \left[\frac{\begin{bmatrix} d \\ n_{\mathsf{E}} \end{bmatrix}}{\begin{bmatrix} L \\ n_{\mathsf{E}} \end{bmatrix}} \mathbb{P}\left[\dim \Pi_{\mathsf{A}} \ge n_{\mathsf{E}}\right] \right]. \tag{8.65}$$

Similar to the previous case, we can apply the Kuhn-Tucker theorem to find the optimal input distribution α_i^*'s.

Taking derivative for $0 \le k < n_{\mathsf{E}}$, we have

$$\frac{\partial f}{\partial \alpha_k} = 0, \tag{8.66}$$

for $k = n_{\mathsf{E}}$ we have

$$\frac{\partial f}{\partial \alpha_k} = \sum_{d=n_{\mathsf{E}}+1}^{n_{\mathsf{A}}} \alpha_d \left[\frac{\log e}{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} \ge n_{\mathsf{E}}\right]} \right] = \frac{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} > n_{\mathsf{E}}\right]}{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} \ge n_{\mathsf{E}}\right]} \log e, \tag{8.67}$$

and finally for $n_{\mathsf{E}} < k \le n_{\mathsf{A}}$ we have

$$\begin{aligned} \frac{\partial f}{\partial \alpha_k} = &- \log \frac{\alpha_k}{\begin{bmatrix} L \\ k \end{bmatrix}} - \log e + \log \left[\frac{\begin{bmatrix} k \\ n_{\mathsf{E}} \end{bmatrix}}{\begin{bmatrix} L \\ n_{\mathsf{E}} \end{bmatrix}} \mathbb{P}\left[\dim \Pi_{\mathsf{A}} \ge n_{\mathsf{E}}\right] \right] \\ &+ \frac{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} > n_{\mathsf{E}}\right]}{\mathbb{P}\left[\dim \Pi_{\mathsf{A}} \ge n_{\mathsf{E}}\right]} \log_2 e. \end{aligned} \tag{8.68}$$

We can easily check that for large q, the input distribution that has $\alpha_{n_{\mathsf{A}}} = 1$ and $\alpha_i = 0$ for $i \neq n_{\mathsf{A}}$ satisfies the Kuhn-Tucker conditions. For this distribution, we have

$$\begin{cases} 0 \leq k < n_{\mathsf{E}} & : \ \frac{\partial f}{\partial \alpha_k} = 0 < \lambda, \\ k = n_{\mathsf{E}} & : \ \frac{\partial f}{\partial \alpha_k} = \log e < \lambda, \\ n_{\mathsf{E}} < k < n_{\mathsf{A}} & : \ \frac{\partial f}{\partial \alpha_k} = \log \begin{bmatrix} k \\ n_{\mathsf{E}} \end{bmatrix} \begin{bmatrix} L \\ n_{\mathsf{E}} \end{bmatrix}^{-1} < \lambda, \\ k = n_{\mathsf{A}} & : \ \frac{\partial f}{\partial \alpha_k} = \log \begin{bmatrix} L \\ n_{\mathsf{A}} \end{bmatrix} \begin{bmatrix} n_{\mathsf{A}} \\ n_{\mathsf{E}} \end{bmatrix} \begin{bmatrix} L \\ n_{\mathsf{E}} \end{bmatrix}^{-1} = \lambda. \end{cases} \tag{8.69}$$

So for the second case, we have

$$\begin{aligned} \max_{P_{\Pi_{\mathsf{A}}}} I(\Pi_{\mathsf{A}}; \Pi_i | \Pi_{\mathsf{E}}) &= \lambda \\ &= \log \begin{bmatrix} L \\ n_{\mathsf{A}} \end{bmatrix} \begin{bmatrix} n_{\mathsf{A}} \\ n_{\mathsf{E}} \end{bmatrix} \begin{bmatrix} L \\ n_{\mathsf{E}} \end{bmatrix}^{-1} \\ &= \log \begin{bmatrix} L - n_{\mathsf{E}} \\ n_{\mathsf{A}} - n_{\mathsf{E}} \end{bmatrix} \\ &\approx (n_{\mathsf{A}} - n_{\mathsf{E}})(L - n_{\mathsf{A}}) \log q. \end{aligned} \tag{8.70}$$

Combining the first and the second case we get the desired result, namely,

$$\max_{P_{\Pi_{\mathsf{A}}}} I(\Pi_{\mathsf{A}}; \Pi_i | \Pi_{\mathsf{E}}) = \big(\min[n_{\mathsf{A}}, n_i + n_{\mathsf{E}}] - n_{\mathsf{E}} \big) \big(L - \min[n_{\mathsf{A}}, n_i + n_{\mathsf{E}}] \big) \log q. \tag{8.71}$$

□

Proof of Lemma 8.5. Let us add $U_{\mathsf{E}} \triangleq \Pi_{\mathsf{E}}$ to $2^m - 1$ subspaces $U_{\mathcal{J}}$'s, where $\emptyset \neq \mathcal{J} \subseteq [1:m]$. Then from the assumption of the lemma, for $k \in [1 : 2^m - 1]$ and any selection of subsets $\mathcal{J}_1, \cdots, \mathcal{J}_k$, we have also

$$\begin{aligned} \theta_{\mathcal{J}_1} + \cdots + \theta_{\mathcal{J}_k} &\leq \dim \left(U_{\mathcal{J}_1} + \cdots + U_{\mathcal{J}_k} + \Pi_{\mathsf{E}} \right) - \dim(\Pi_{\mathsf{E}}) \\ &\leq \dim \left(U_{\mathcal{J}_1} + \cdots + U_{\mathcal{J}_k} \right). \end{aligned} \tag{8.72}$$

Now by defining $\theta_{\mathsf{E}} \triangleq \dim(\Pi_{\mathsf{E}})$, we can apply [101, Lemma 4] to the set of subspaces $U_{\mathcal{J}}$'s and U_{E} to show that there exist subspaces $U'_{\mathcal{J}} \sqsubseteq U_{\mathcal{J}}$ such that $\dim(U'_{\mathcal{J}}) = \theta_{\mathcal{J}}$ for $\emptyset \neq \mathcal{J} \subseteq [1 : m]$, and $U'_{\mathsf{E}} = \Pi_{\mathsf{E}}$ where all of them are complementary. Note that in the above argument, we have $U'_{\mathsf{E}} = \Pi_{\mathsf{E}}$ because we set $\theta_{\mathsf{E}} = \dim(\Pi_{\mathsf{E}})$ (which is an integer number). □

Proof of Lemma 8.6. Let us assume that Alice has Π_{E}. Then she can create subspaces $U'_{\mathcal{J}}$'s such that by using Lemma 8.5, for $k \in [1 : 2^m - 1]$ and any selection of subsets $\mathcal{J}_1, \ldots, \mathcal{J}_k$, we have

$$\dim(U'_{\mathcal{J}_1} + \cdots + U'_{\mathcal{J}_k} + \Pi_{\mathsf{E}}) = \theta_{\mathcal{J}_1} + \cdots + \theta_{\mathcal{J}_k} + \dim(\Pi_{\mathsf{E}}), \tag{8.73}$$

which means that

$$\theta_{\mathcal{J}_1} + \cdots + \theta_{\mathcal{J}_k} + \dim(\Pi_{\mathsf{E}}) \leq n_{\mathsf{A}}. \tag{8.74}$$

Now, suppose that Alice does not have access to Π_{E}. From the proof of Lemma 8.5, we know that for any k and any subsets $\mathcal{J}_1, \ldots, \mathcal{J}_k$ we have also

$$\theta_{\mathcal{J}_1} + \cdots + \theta_{\mathcal{J}_k} \leq \dim\left(U_{\mathcal{J}_1} + \cdots + U_{\mathcal{J}_k}\right), \tag{8.75}$$

so by using [101, Lemma 4] Alice can find subspaces $U'_{\mathcal{J}}$'s such that they are complementary and $\dim(U'_{\mathcal{J}}) = \theta_{\mathcal{J}}$ for every subset $\mathcal{J}$.

From Alice's point of view Π_{E}, is chosen independently and uniformly at random from Π_{A}. So by (8.74) and applying Lemma 2.11, the subspace Π_{E} is complementary to all $U'_{\mathcal{J}}$'s w.h.p and we are done. □

Proof of Lemma 8.7. Construct matrix $\boldsymbol{Y}$ that has as rows the packets $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k$. Then note that we can write

$$\begin{bmatrix} \boldsymbol{Y} \\ \boldsymbol{X}_{\mathsf{E}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{F}_{\mathsf{E}} \end{bmatrix} \boldsymbol{X}_{\mathsf{A}} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{F}_{\mathsf{E}} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{M} \end{bmatrix}, \tag{8.76}$$

where $\boldsymbol{M}$ is a uniformly random matrix and $\boldsymbol{A} \in \mathbb{F}_q^{k \times n_{\mathsf{A}}}$ is the coefficients of $\boldsymbol{Y}$ packets.

We now proceed by expanding $H(\boldsymbol{Y}|\boldsymbol{X}_{\mathsf{E}})$. We have

$$\begin{aligned} H(\boldsymbol{Y}|\boldsymbol{X}_{\mathsf{E}}) &= H(\boldsymbol{Y}, \boldsymbol{X}_{\mathsf{E}}) - H(\boldsymbol{X}_{\mathsf{E}}) \\ &= H(\boldsymbol{AM}, \boldsymbol{F}_{\mathsf{E}}\boldsymbol{M}) - H(\boldsymbol{F}_{\mathsf{E}}\boldsymbol{M}) \\ &= \big[\operatorname{rank}(\boldsymbol{B}) - \operatorname{rank}(\boldsymbol{F}_{\mathsf{E}})\big](L - n_{\mathsf{A}}) \log q, \end{aligned} \tag{8.77}$$

where

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{F}_{\mathsf{E}} \end{bmatrix} \in \mathbb{F}_q^{(k+n_{\mathsf{E}}) \times n_{\mathsf{A}}}. \tag{8.78}$$

Because $\Pi_Y = \langle \boldsymbol{Y} \rangle$ is orthogonal to $\langle \boldsymbol{X}_{\mathsf{E}} \rangle$ we have that $\langle \boldsymbol{A} \rangle$ is orthogonal to $\langle \boldsymbol{F}_{\mathsf{E}} \rangle$. Thus we can write

$$\operatorname{rank}(\boldsymbol{B}) = \operatorname{rank}(\boldsymbol{A}) + \operatorname{rank}(\boldsymbol{F}_{\mathsf{E}}). \tag{8.79}$$

Finally, we can proceed as follows

$$\begin{aligned} I(\boldsymbol{Y}; \boldsymbol{X}_{\mathsf{E}}) &= H(\boldsymbol{Y}) - H(\boldsymbol{Y}|\boldsymbol{X}_{\mathsf{E}}) \\ &= \big[\operatorname{rank}(\boldsymbol{A}) - [\operatorname{rank}(\boldsymbol{B}) - \operatorname{rank}(\boldsymbol{X}_{\mathsf{E}})]\big](L - n_{\mathsf{A}}) \log q \\ &= 0. \end{aligned} \tag{8.80}$$

This proves the assertion of the lemma. □

"To achieve great things, two things are needed; a plan, and not quite enough time."

Leonard Bernstein

9 Discussion and Future Directions

In this thesis, we have mainly considered communication networks which allow for inter-network linear operations, i.e., the relay nodes perform linear NC. Although a complete theory of network information is yet to be discovered, the advent of NC paradigm opens new opportunities and provides new insights for network information flow algorithms. During the thesis, we have focused on various problems that arise in this context.

Non-coherent Network Coding

In the first part of the thesis, we have studied a non-coherent NC scenario, motivated by randomized NC, where neither source(s) nor destination(s) do not have any knowledge about the network operation. To this end, we have considered two different channel models where in both cases the network operation is captured by a multiplicative matrix channel over some finite field.

In the first model, we have assumed that the transfer matrix is distributed uniformly among all possible matrices. For this communication scenario, we have characterized the capacity for a unicast communication as well as we have derived the rate region of a two users multiple access scenario.[1] Our results shows that coding over subspaces is sufficient to achieve the capacity. Moreover, we have proved that the throughput benefits subspace coding offers as compared to the use of coding vectors go to zero as the alphabet size q increases, and thus use of coding vectors is (asymptotically) optimal. However, when the alphabet size is small finding an efficient coding scheme is still an open problem.

1. It is worthwhile to mention that while our MAC result is only derived for two users, the same technique can be extended to find the rate region of a non-coherent NC MAC problem with more than two users.

In the second model, which is in some sense more universal than the previous one, the network operation is captured by a multiplicative matrix channel, however with only a known distribution over the rank of the transfer matrix. This model recalls the classical AVC problem but with probabilistic constraints as opposed to deterministic constraints that have been considered in the literature. We have named such an AVC problem with probabilistic constraints as partially AVC (PAVC). For this model of non-coherent NC, we have characterized the capacity as well and prove that subspace coding is again sufficient to achieve it. However, finding an efficient coding scheme to achieve the capacity for the cases where the rank distribution of the transfer matrix is not concentrated over some number is an open problem.

In all of the previous models, we have not taken into account the correlation between the network topology and the channel model of the non-coherent NC, i.e., captured by the distribution of the transfer matrix. This has been done because the exact characterization of the transfer matrix based on the network topology is an extremely hard problem. However, as a future direction of research, even finding an approximate relation between network topology and the transfer matrix's distribution would greatly help us to analyze more realistic and practical models.

Subspace Properties of Network Coding

We have started the second part of the thesis by observing that the packets (message vectors) traversing the networks carry topological and state-dependent information about the network. In order to distill this information, we have investigated the properties of subspaces spanned by the packets received at every node [2]. To this end, we have studied the properties of randomly selected subspaces from a linear space defined over a finite field in Chapter 2. Then, these properties have been adapted in Chapter 6 towards different applications. As the first application, by extending the aforementioned properties to random subspaces evolving over time, we have studied the conditions under which we can passively infer the network topology during content dissemination by having access to a global view of the network. As the second application, which in some sense is the dual of the previous problem, we have focused on locating Byzantine attackers in the network. Finally, in the last application, we have observed that even having access to the subspaces received at a node, we can obtain some information regarding the network. So we have leveraged this observation and proposed a decentralized scheme for breaking bottlenecks in the network.

2. Note that in case of using coding vectors, the subspaces spanned by the coding vectors is isomorphic to the subspaces spanned by the whole message vectors, as explained in Chapter 6. So instead of considering the subspaces spanned by the message vectors we may only focus on the subspaces spanned by the coding vectors.

Secrecy

In the last part of the thesis, we have focused on the secret key sharing problem among multiple terminals from an information theoretical point of view. More precisely, we have studied the problem of secret key sharing among multiple trusted (authenticated) entities having access to a broadcast channel which is overheard by a passive eavesdropper. In addition to the broadcast channel, the trusted terminals can discuss over a public channel. For the above setup, we have been interested in characterizing the secrecy capacity in non-coherent NC scenarios as well as wireless environments.

Although it seems that these two problems differ very much in nature, we have used similar techniques based on the insights we gained from studying the erasure broadcast channel. The proposed achievability scheme for the erasure broadcast channel achieves the secrecy capacity efficiently (it is a polynomial time algorithm) and it is based on ideas from NC to reconciliate the secret key among the terminals. Then, by extending this achievability scheme we have proposed schemes for secret key sharing among multiple terminals for non-coherent broadcast channels as well as state-dependent Gaussian broadcast channels.

There are many open questions in this context. The above-mentioned problem of finding the capacity of secret key sharing among multiple terminals in its general form (for an arbitrary broadcast channel) is still open. Even the secrecy capacity for the non-coherent and state-dependent Gaussian broadcast channels is unknown. One of the most important questions for this problem (in its general form) is that whether or not it is sufficient to achieve the secrecy capacity by converting the (channel) problem to a source problem where Alice emulates a multi-terminal correlated source by sending a random sequence over the broadcast channel.

Partial List of Symbols

In the following, a list of frequently used notations, symbols, and abbreviations are presented.

$\triangleq$	Definition.
$\mathbb{N}$	Set of natural numbers.
$\mathbb{Z}$	Set of integer numbers.
$\mathbb{R}$	Set of real numbers.
$\mathbb{R}_{+}$	Set of non-negative real numbers.
$\mathbb{R}_{++}$	Set of positive real numbers.
$[i:j]$	Set of integer numbers $\{i, i+1, \ldots, j\}$ where $i, j \in \mathbb{Z}$ and $i \leq j$.
$\mathbb{F}_q$	Finite field of size q.
$\mathbb{F}_q^{m \times n}$	The set of all $m \times n$ matrices over $\mathbb{F}_q$.
$\mathbb{F}_q^{m \times n, k}$	The set of all $m \times n$ matrices over $\mathbb{F}_q$ with rank k.
$\mathrm{hwt}(\mathbf{v})$	Hamming weight of a vector $\mathbf{v}$.
$\succ$ and $\prec$	Element-wise inequality between vectors and matrices of the same size.
$\subseteq$	Subset relation.
$\sqsubseteq$	Subspace relation.
$[n, k, d]_q$	A linear code of length n, dimension k, and minimum distance d defined over $\mathbb{F}_q$.
$\begin{bmatrix} n \\ m \end{bmatrix}_q$	The Gaussian number; see Definition 2.2.
$\mathsf{Uni}(\mathcal{M})$	Uniform distribution over a set $\mathcal{M}$.
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2.
$\mathbb{P}[\mathcal{A}]$	Probability of an event $\mathcal{A}$.
$\mathbb{E}[\cdot]$	Expectation operator.
$\mathrm{var}(X)$	The variance of a random variable X.
$\mathrm{cov}(\mathbf{v})$	The covariance matrix of a random vector $\mathbf{v}$.
AVC	Arbitrarily varying channel.
CSI	Channel state information.
DMC	Discrete memory-less channel.
GLFP	Generalized linear fractional programming.

MAC	Multiple access channel.
MDS	Maximum distance separable.
MIMO	Multiple-input and multiple-output.
NC	Network coding.
P2P	Peer-to-Peer.
PAVC	Partially arbitrarily varying channel. See Chapter 4 for the definition.
SNR	Signal to noise ratio.

Bibliography

[1] R. Koetter and F. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3579–3591, Aug. 2008.

[2] "Visual networking index," cisco Systems, [Online]. Available: http://www.cisco.com/go/vni.

[3] "Minnesota internet traffic studies (mints)," university of Minnesota, [Online]. Available: http://www.dtc.umn.edu/mints/home.php.

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

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

[6] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204–1216, Jul. 2000.

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

[8] C. Fragouli and E. Soljanin, "Network coding fundamentals," in *Monograph in Series, Foundations and Trends in Networking*. Now Publishers, Jun. 2007.

[9] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for network information flow," *ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pp. 286–294, 2003.

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

[11] N. J. A. Harvey, D. R. Karger, and K. Murota, "Deterministic network coding by matrix completion," *ACM-SIAM Aymposium on Discrete Algorithms (SODA)*, pp. 489–498, 2005.

[12] A. I. Barbero and O. Ytrehus, "Heuristic algorithms for small field multicast encoding," *IEEE Information Theory Workshop (ITW)*, pp. 428–432, Oct. 2006.

[13] J. H. van Lint and R. M. Wilson, *A course in combinatorics*, 2nd ed. Cambridge University Press, 2001.

[14] G. Andrews, *The theory of partitions*. Encyclopedia of Mathematics and its Applications, 1976.

[15] M. Gadouleau and Z. Yan, "On the decoder error probability of bounded rank-distance decoders for maximum rank distance codes," *IEEE Transactions on Information Theory*, vol. 54, no. 7, pp. 3202–3206, Jul. 2008.

[16] E. Gabidulin, "Theory of codes with maximum rank distance," *Problems of Information Transmission*, vol. 21, no. 1, pp. 1–12, Jan. 1985.

[17] D. Laksov and A. Thorup, "Counting matrices with coordinates in finite fields and of fixed rank," *Mathematica Scandinavica*, vol. 74, pp. 19–33, 1994.

[18] M. Gadouleau and Z. Yan, "Packing and covering properties of subspace codes for error control in random linear network coding," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2097–2108, May 2010.

[19] L. Babai and P. Frankl, "Linear algebra methods in combinatorics," preliminary version, University of Chicago.

[20] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," *Allerton Conference on Communication, Control, and Computing*, Oct. 2003.

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

[22] TinyOs. http://www.tinyos.net/.

[23] L. Keller, M. Jafari Siavoshani, C. Fragouli, K. Argyraki, and S. N. Diggavi, "Joint identity-message coding," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, pp. 1083–1093, Sep. 2010.

[24] M. Jafari Siavoshani, C. Fragouli, and S. N. Diggavi, "Non-coherent multisource network coding," *IEEE International Symposium on Information Theory (ISIT)*, pp. 817–821, Jul. 2008.

[25] A. Montanari and R. Urbanke, "Coding for network coding," 2007, [Online]. Available: http://arxiv.org/abs/0711.3935/.

[26] D. Silva, F. R. Kschischang, and R. Koetter, "Communication over finite-field matrix channels," *IEEE Transactions on Information Theory*, vol. 56, no. 3, pp. 1296–1305, Mar. 2010.

[27] M. Jafari Siavoshani, S. Mohajer, C. Fragouli, and S. N. Diggavi, "On the capacity of non-coherent network coding," *IEEE International Symposium on Information Theory (ISIT)*, pp. 273–277, Jun. 2009.

[28] S. Yang, S.-W. Ho, J. Meng, E. hui Yang, and R. W. Yeung, "On linear operator channels over finite fields," 2010, [Online]. Available: http://arxiv.org/abs/1002.2293v2.

[29] S. Yang, S.-W. Ho, J. Meng, and E. hui Yang, "Symmetric properties and subspace degradations of linear operator channels over finite fields," 2011, [Online]. Available: http://arxiv.org/abs/1108.4257.

[30] S. Yang, J. Meng, and E. hui Yang, "Coding for linear operator channels over finite fields," *IEEE International Symposium on Information Theory (ISIT)*, pp. 2413–2417, Jun. 2010.

[31] S. Yang, S.-W. Ho, J. Meng, and E. hui Yang, "Optimality of subspace coding for linear operator channels over finite fields," *IEEE Information Theory Workshop (ITW)*, pp. 1–5, Jan. 2010.

[32] R. W. Nobrega, B. F. Uchoa-Filho, and D. Silva, "On the capacity of multiplicative finite-field matrix channels," *IEEE International Symposium on Information Theory (ISIT)*, pp. 341–345, Jul.-Aug. 2011.

[33] M. Jafari Siavoshani, S. Mohajer, C. Fragouli, and S. N. Diggavi, "On the capacity of non-coherent network coding," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 1046–1066, Feb. 2011.

[34] L. Zheng and D. N. C. Tse, "Communication on the grassmannian manifold: A geometric approach to the non-coherent multiple-antenna channel," *IEEE Transactions on Information Theory*, vol. 48, no. 2, pp. 359–383, Feb. 2002.

[35] M. Jafari Siavoshani, C. Fragouli, , and S. N. Diggavi, "Subspace properties of randomized network coding," *IEEE Information Theory Workshop (ITW)*, pp. 1–5, Jul. 2007.

[36] P. Sattari, A. Markopoulou, and C. Fragouli, "Multiple source multiple destination topology inference using network coding," *Workshop on Network Coding, Theory, and Applications (NetCod)*, pp. 36–41, Jun. 2009.

[37] G. Sharma, S. Jaggi, and B. K. Dey, "Network tomography via network coding," *Information Theory and Application Workshop (ITA)*, pp. 151–157, Feb. 2008.

[38] T. Cover and J. Thomas, *Elements of Information Theory*, 2nd ed. Wiley and Sons, 2006.

[39] K. Price and R. Storn, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[40] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[41] D. Silva, F. Kschischang, and R. Koetter, "A rank-metric approach to error control in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 9, pp. 3951–3967, Sep. 2008.

[42] S. Mohajer, M. Jafari Siavoshani, S. N. Diggavi, and C. Fragouli, "On the capacity of multisource non-coherent network coding," *IEEE Information Theory Workshop (ITW)*, pp. 130–134, Jun. 2009.

[43] D. Blackwell, L. Breiman, and A. J. Thomasian, "The capacities of certain channel classes under random coding," *The Annals of Mathematical Statistics*, vol. 31, no. 2, pp. 558–567, Sep. 1960.

[44] A. Lapidoth and P. Narayan, "Reliable communication under channel uncertainty," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2148–2177, Oct. 1998.

[45] I. Csiszar and P. Narayan, "The capacity of the arbitrarily varying channel revisited: Positivity, constraints," *IEEE Transactions on Information Theory*, vol. 34, no. 1, pp. 181–193, Jan. 1988.

[46] I. Csiszar and J. Korner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Academic Press, 1981.

[47] I. Csiszar and P. Narayan, "Arbitrarily varying channels with constrained inputs and states," *IEEE Transactions on Information Theory*, vol. 34, no. 1, pp. 27–34, Jan. 1988.

[48] F. J. Macwilliams and N. J. A. Sloane, *The theory of error correcting codes*, 2nd ed. North-Holland Mathematical Library, 1978.

[49] E. R. Berlekamp, R. J. McEliece, and H. C. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 384–386, May 1978.

[50] V. S. Pless and W. C. Huffman, *Handbook of coding theory*. North-Holland Mathematical Library, 1998.

[51] I. S. Reed and G. Solomon, "Polynomial codes over certain finite field," *Journal of the Society for Industrial and Applied Mathematics (SIAM)*, vol. 8, pp. 300–304, 1960.

[52] V. D. Goppa, "A new class of linear error-correcting codes," *Probl. Peredach. Inform.*, vol. 6, no. 3, pp. 24–30, 1970.

[53] ——, "Codes associated with divisors," *Probl. Peredachi Inform.*, vol. 13, pp. 33–39, 1997, translation: *Probl. Inform. Transmission*, vol. 13, pp. 22–26, 1977.

[54] E. R. Berlekamp, "Nonbinary bch decoding," *IEEE Transactions on Information Theory*, vol. 14, no. 2, p. 242, Mar. 1968.

[55] J. L. Massey, "Shift-register synthesis and bch decoding," *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 122–127, Jan. 1969.

[56] R. E. Blahut, *Algebraic Codes for Data Transmission*. Cambridge University Press, 2003.

[57] M. Grassl, "Bounds on the minimum distance of linear codes," Jan. 2009, [Online]. Available: http://www.codetables.de, Accessed on 2009-01-09.

[58] J. Blomer, R. Karp, and E. Welzl, "The rank of sparse random matrices over finite fields," *Random Structures and Algorithms*, vol. 10, no. 4, pp. 407–419, Jul. 1997.

[59] P. Pakzad, C. Fragouli, and A. Shokrollahi, "Coding schemes for line networks," *IEEE International Symposium on Information Theory (ISIT)*, pp. 1853–1857, Sep. 2005.

[60] L. Keller, M. Jafari Siavoshani, K. Argyraki, C. Fragouli, and S. Diggavi, "Identity aware sensor networks," *IEEE International Conference on Computer Communications (INFOCOM)*, pp. 2177–2185, Apr. 2009.

[61] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," *IEEE International Conference on Computer Communications (INFOCOM)*, vol. 4, pp. 2235–2245, Mar. 2005.

[62] C. Gkantsidis, J. Miller, and P. R. Rodriguez, "Comprehensive view of a live network coding p2p system," *ACM SIGCOMM conference on Internet measurement (IMC)*, pp. 177–188, 2006.

[63] C. Fragouli, J. Widmer, and J. Y. L. Boudec, "A network coding approach to energy efficient broadcasting: from theory to practice," *IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1–11, Apr. 2006.

[64] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," *IEEE International Symposium on Information Theory (ISIT)*, p. 442, Jun. Jul. 2003.

[65] W. Li-shuang, S. Wei, H. Wen-bin, and H. Zheng-bing, "Using network coding make p2p content sharing scalable," *International Workshop on Database Technology and Applications (DBTA)*, pp. 1–4, Nov. 2010.

[66] X. Wei and D.-Y. Long, "P2p content-propagation mechanism tailored by network coding," *International Symposium on Computer Network and Multimedia Technology (CNMT)*, pp. 1–6, Jan. 2009.

[67] X. Zhang and B. Li, "On the market power of network coding in p2p content distribution systems," *IEEE International Conference on Computer Communications (INFOCOM)*, pp. 334–342, Apr. 2009.

[68] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Medard, "Resilient network coding in the presence of byzantine adversaries," *IEEE International Conference on Computer Communications (INFOCOM)*, pp. 616–624, May 2007.

[69] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros, and D. Karger, "Byzantine modification detection in multicast networks using randomized network coding," *IEEE International Symposium on Information Theory (ISIT)*, p. 144, Jun. Jul. 2004.

[70] R. W. Yeung and N. Cai, "Network error correction, i: basic concepts and upper bounds," *Communication and Information System*, vol. 6, pp. 19–35, 2006.

[71] N. Cai and R. W. Yeung, "Network error correction, ii: lower bounds," *Communication and Information System*, vol. 6, pp. 37–54, 2006.

[72] Z. Zhang, "Network error correction coding in packetized networks," *IEEE Information Theory Workshop (ITW)*, pp. 433–437, Oct. 2006.

[73] D. Silva and F. R. Kschischang, "Universal secure network coding via rank-metric codes," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 1124–1135, Feb. 2011.

[74] M. Jafari Siavoshani, C. Fragouli, and S. N. Diggavi, "On locating byzantine attackers," *Workshop on Network Coding, Theory, and Applications (NetCod)*, pp. 1–6, Jan. 2008.

[75] E. Kehdi and B. Li, "Null keys: Limiting malicious attacks via null space properties of network coding," *IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1224–1232, Apr. 2009.

[76] RON: Resilient Overlay Networks, [Online]. Available: http://nms.csail.mit.edu/ron.

[77] C. Fragouli and A. Markopoulou, "A network coding approach to overlay network monitoring," *Allerton Conference on Communication, Control, and Computing*, Sep. 2005.

[78] C. Fragouli, A. Markopoulou, and S. N. Diggavi, "Active topology inference using network coding," *Allerton Conference on Communication, Control, and Computing*, Sep. 2006.

[79] T. Ho, B. Leong, Y. Chang, Y. Wen, and R. Koetter, "Network monitoring in multicast networks using network coding," *IEEE International Symposium on Information Theory (ISIT)*, pp. 1977–1981, Sep. 2005.

[80] D. S. Lun, N. Ratnakar, M. Medard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost multicast over coded packet networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2608–2623, Jun. 2006.

[81] A. Al-Hamra, A. Legout, and C. Barakat, "Understanding the properties of the bittorrent overlay," *INRIA Technical Report*, 2007, [Online]. Available: http://arxiv.org/pdf/0707.1820.

[82] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge University Press, 2000.

[83] R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography, part i: secret sharing," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1121–1132, Jul. 1993.

[84] U. M. Maurer, "Secret key agreement by public discussion from common information," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 733–742, May 1993.

[85] I. Csiszar and P. Narayan, "Secrecy capacities for multiterminal channel models," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2437–2452, Jun. 2008.

[86] A. A. Gohari and V. Anantharam, "Information-theoretic key agreement of multiple terminals - part ii: Channel model," *IEEE Transactions on Information Theory*, vol. 56, no. 8, pp. 3997–4010, Aug. 2010.

[87] A. Khisti, S. N. Diggavi, and G. W. Wornell, "Secret-key agreement over wiretap channels with random state parameters," *IEEE Transactions on Information Forensics and Security*, 2011.

[88] A. S. Avestimehr, S. N. Diggavi, and D. N. C. Tse, "Wireless network information flow," *Allerton Conference on Communication, Control, and Computing*, Sep. 2007.

[89] ——, "Wireless network information flow: A deterministic approach," *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 1872–1905, Apr. 2011.

[90] Y. Liang, L. Lai, H. V. Poor, and S. S. (Shitz), "The broadcast approach over fading gaussian wiretap channels," *IEEE Information Theory Workshop (ITW)*, pp. 1–5, Oct. 2009.

[91] A. D. Wyner, "The wire-tap channel," *Bell System Tech. J.*, vol. 54, pp. 1355–1387, Oct. 1975.

[92] A. A. Gohari and V. Anantharam, "Information-theoretic key agreement of multiple terminals - part i," *IEEE Transactions on Information Theory*, vol. 56, no. 8, pp. 3973–3996, Aug. 2010.

[93] C. Chan, "Generating secret in a network," Ph.D. dissertation, Deptartement of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2010.

[94] D. Bertsekas, A. Nedic, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Athena Scientific, 2003.

[95] N. T. H. Phuong and H. Tuy, "A unified monotonic approach to generalized linear fractional programming," *Journal of Global Optimization*, vol. 26, pp. 229–259, 2003.

[96] L. P. Qian, Y. J. Zhang, and J. Huang, "Mapel: Achieving global optimality for a non-convex wireless power control problem," *IEEE Transactions on Communication*, vol. 8, no. 3, pp. 1553–1563, Mar. 2009.

[97] M. Mitzenmacher and E. Upfal, *Probability and computing, randomized algorithm and probabilistic analysis*. Cambridge University Press, 2006.

[98] N. Cai and R. W. Yeung, "Secure network coding," *IEEE International Symposium on Information Theory (ISIT)*, p. 323, 2002.

[99] J. Feldman, T. Malkin, C. Stein, and R. A. Servedio, "On the capacity of secure network coding," *Allerton Conference on Communication, Control, and Computing*, Sep. 2004.

[100] S. Y. E. Rouayheb and E. Soljanin, "On wiretap networks ii," *IEEE International Symposium on Information Theory (ISIT)*, pp. 551–555, Jun. 2007.

[101] M. A. Khojastepour and A. Keshavarz-Haddad, "Multicast achievable rate region of deterministic broadcast channel," *IEEE International Conference on Communications (ICC)*, 2011.

[102] Y. Liang, H. V. Poor, and S. S. (Shitz), "Information theoretic security," *Foundations and Trends in Communications and Information Theory*, vol. 5, no. 4–5, pp. 355–580, 2009.

[103] I. Csiszar and J. Korner, "Broadcast channels with confidential messages," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 339–348, May 1978.

Curriculum Vitae

Mahdi Jafari Siavoshani

Address: EPFL - I&C - IIF - ARNI
BC 046 (Bâtiment BC), Station 14
CH-1015 Lausanne, Switzerland
Phone: +41(21)693-1354
Fax: +41(21)693-1200

Email: mahdi.jafarisiavoshani@epfl.ch
Web: http://arni.epfl.ch/~jafari

Education

2007–2012 **Ph.D. in Computer and Communication Sciences**
Ecole Polytechnique Fédérale de Lausanne, Switzerland
Dissertation Title: *Network Coding: Theoretical Designs Directed to Applications*
Supervisor: *Prof. Christina Fragouli*

2005–2007 **M.Sc. in Computer and Communication Sciences**
Ecole Polytechnique Fédérale de Lausanne, Switzerland
Dissertation Title: *On Randomized Network Coding Properties and their Applications*
Supervisor: *Prof. Christina Fragouli*

2000–2004 **B.Sc. in Communication Systems**
Sharif University of Technology, Tehran, Iran
Dissertation Title: *Application of Radio over Fiber in Cellular Communication*
Supervisor: *Prof. Jawad A. Salehi*

2003–2005 **B.Sc. in Physics**
Sharif University of Technology, Tehran, Iran

Publications

Working Papers

- M. Jafari Siavoshani, S. Yang, and R. W. Yeung, "Non-coherent network coding: An arbitrarily varying channel approach."
- M. Jafari Siavoshani, S. Mishra, C. Fragouli, and S. N. Diggavi, "Group secret key agreement over state-dependent wireless broadcast channels."

Journal Papers

- M. Jafari Siavoshani, C. Fragouli, and S. Diggavi, "Subspace properties of network coding and their applications," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 2599–2619, May 2012.
- M. Jafari Siavoshani, S. Mohajer, C. Fragouli, and S. Diggavi, "On the capacity of non-coherent network coding," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 1046–1066, Feb. 2011.
- L. Keller, M. Jafari Siavoshani, K. Argyraki, C. Fragouli, and S. Diggavi, "Joint identity-message coding," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 28, no. 7, Sep. 2010.

Conference Proceedings

- M. Jafari Siavoshani, S. Yang, and R. W. Yeung, "Non-coherent network coding: an arbitrarily varying channel approach," *IEEE International Symposium on Information Theory (ISIT)*, U.S.A., Jul. 2012.
- M. Jafari Siavoshani, C. Fragouli, "Multi-terminal Secrecy in Linear Non-coherent Packetized Networks," *International Symposium on Network Coding (NETCOD)*, U.S.A., Jun. 2012.
- M. Jafari Siavoshani, S. Mishra, C. Fragouli, and S. N. Diggavi, "Group secret key agreement over state-dependent wireless broadcast channels," *IEEE International Symposium on Information Theory (ISIT)*, Russia, Aug. 2011.
- M. Jafari Siavoshani, C. Fragouli, S. Diggavi, U. Pulleti, and K. Argyraki, "Group secret key generation over broadcast erasure channels," *Asilomar Conference on Signals, Systems, and Computers*, Nov. 2010.
- S. Mohajer, M. Jafari Siavoshani, S. N. Diggavi, and C. Fragouli, "On the capacity of multisource non-coherent network coding," *Information Theory Workshop (ITW)*, Jun. 2009.
- M. Jafari Siavoshani, C. Fragouli, and S. Diggavi, "Code construction for multiple sources network coding," MobiHoc-S3'09, May 2009.

- M. Jafari Siavoshani, L. Keller, C. Fragouli, and K. Argyraki, "Compressed network coding vectors," *IEEE International Symposium on Information Theory (ISIT)*, Jun. 2009.
- M. Jafari Siavoshani, S. Mohajer, C. Fragouli, and S. Diggavi, "On the capacity of non-coherent network coding," *IEEE International Symposium on Information Theory (ISIT)*, Jun. 2009.
- L. Keller, M. Jafari Siavoshani, K. Argyraki, C. Fragouli, and S. Diggavi, "Identity aware sensor networks," *IEEE INFOCOM*, Apr. 2009.
- M. Jafari Siavoshani, C. Fragouli, and S. Diggavi, "Noncoherent multisource network coding," *IEEE International Symposium on Information Theory (ISIT)*, Jul. 2008.
- M. Jafari Siavoshani, C. Fragouli, and S. Diggavi, "On locating byzantine attackers," *Network Coding Workshop (NETCOD)*, Jan. 2008.
- M. Jafari Siavoshani, C. Fragouli, and S. Diggavi, "Subspace properties of randomized network coding," *IEEE Information Theory Workshop (ITW)*, Jul. 2007.
- M. Jafari Siavoshani, C. Fragouli, S. Diggavi, and C. Gkantsidis, "Bottleneck discovery and overlay management in network coded peer-to-peer systems," *SIGCOMM Workshop on Internet Network Management*, Aug. 2007.

References

1. **Prof. Christina Fragouli:** School of Computer and Communication Sciences (I&C), Ecole Polytechnique Fédérale de Lausanne, Switzerland
   email: *christina.fragouli@epfl.ch*
   phone: +41(21)693-7513
2. **Prof. Suhas Diggavi:** School of Electrical Engineering (EE), University of California, Los Angeles (UCLA), USA
   email: *suhasdiggavi@ucla.edu*
   phone: +1(310)206-5171
3. **Prof. Raymond Yeung:** Department of Information Engineering, The Chinese University of Hong Kong (CUHK), Hong Kong
   email: *whyeung@ie.cuhk.edu.hk*
   phone: +(852)3943-8375
4. **Prof. Jawad A. Salehi:** Electrical Engineering Department (EE), Sharif University of Technology (SUT), Tehran, Iran
   email: *jasalehi@sharif.edu*
   phone: +98(21)6616-4346