

NETWORK AND INDEX CODING
WITH APPLICATIONS TO ROBUST AND SECURE COMMUNICATIONS

A Dissertation

by

SALIM YAACOUB EL ROUAYHEB

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2009

Major Subject: Electrical Engineering

NETWORK AND INDEX CODING
WITH APPLICATIONS TO ROBUST AND SECURE COMMUNICATIONS

A Dissertation

by

SALIM YAACOUB EL ROUAYHEB

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Co-Chairs of Committee,	Costas N. Georghiades Alexander Sprintson
Committee Members,	Aniruddha Datta Jean-Francois Chamberland J. Maurice Rojas
Head of Department,	Costas N. Georghiades

December 2009

Major Subject: Electrical Engineering

ABSTRACT

Network and Index Coding

with Applications to Robust and Secure Communications. (December 2009)

Salim Yaacoub El Rouayheb, B.En., Lebanese University;

M.En., American University Of Beirut

Co-Chairs of Advisory Committee: Dr. Costas N. Georghiades
Dr. Alexander Sprintson

Since its introduction in the year 2000 by Ahlswede et al., the *network coding* paradigm has revolutionized the way we understand information flows in networks. Traditionally, information transmitted in a communication network was treated as a commodity in a transportation network, much like cars on highways or fluids in pipes. This approach, however, fails to capture the very nature of information, which in contrast to material goods, can be coded and decoded. The network coding techniques take full advantage of the inherent properties of information, and allow the nodes in a network, not only to store and forward, but also to “mix”, i.e., encode, their received data. This approach was shown to result in a substantial throughput gain over the traditional routing and tree packing techniques.

In this dissertation, we study applications of network coding for guarantying *reliable* and *secure* information transmission in networks with compromised edges. First, we investigate the construction of robust network codes for achieving network resilience against link failures. We focus on the practical important case of unicast networks with non-uniform edge capacities where a single link can fail at a time. We

demonstrate that these networks exhibit unique structural properties when they are minimal, i.e., when they do not contain redundant edges. Based on this structure, we prove that robust linear network codes exist for these networks over $GF(2)$, and devise an efficient algorithm to construct them.

Second, we consider the problem of securing a multicast network against an eavesdropper that can intercept the packets on a limited number of network links. We recast this problem as a network generalization of the classical wiretap channel of Type II introduced by Ozarow and Wyner in 1984. In particular, we demonstrate that perfect secrecy can be achieved by using the Ozarow-Wyner scheme of *coset coding* at the source, on top of the implemented network code. Consequently, we transparently recover important results available in the literature on secure network coding. We also derive new bounds on the required secure code alphabet size and an algorithm for code construction.

In the last part of this dissertation, we study the connection between *index coding*, network coding, and matroid linear representation. We devise a reduction from the index coding problem to the network coding problem, implying that in the linear case these two problems are equivalent. We also present a second reduction from the matroid linear representability problem to index coding, and therefore, to network coding. The latter reduction establishes a strong connection between matroid theory and network coding theory. These two reductions are then used to construct special instances of the index coding problem where vector linear codes outperform scalar linear ones, and where non-linear encoding is needed to achieve the optimal number of transmission. Thereby, we provide a counterexample to a related conjecture in the literature and demonstrate the benefits of vector linear codes.

To my parents

ACKNOWLEDGMENTS

I wish to express my gratitude to my advisor, Dr. Costas Georghiades, for his continued support and guidance throughout my doctoral studies. I greatly benefited from his openness to explore new problems and his constant encouragement to pursue my own research interests. The questions and ideas he brought up in our meetings contributed largely to deepening my understanding of my research and many times steered it into new fruitful directions. I was also fortunate to have Dr. Alex Sprintson as a second advisor. I would like to thank him for always being ready for a meeting to discuss and enrich my ideas. His help, trust and determination were crucial factors in keeping alive within me the enthusiasm that led to bringing to light many of the results presented in this dissertation. I would also like to thank the remaining members of my advisory committee: Dr. Jean-Francois Chamberland, Dr. Aniruddha Datta and Dr. J. Maurice Rojas. My thanks are also due to Paula Evans for her daily assistance at the Wireless Communication Lab.

Very special thanks go to Dr. Emina Soljanin who gave me the great opportunity of working with her as an intern at the Mathematics of Communication Research Department at Bell Labs during the summer of 2006. Her genuine care for students, in conjunction with her friendly and humane mentoring skills, resulted in a fruitful collaboration and a very enjoyable internship experience. I would like also to take this opportunity to thank Prof. Joaquim Hagenauer and my friend, Christoph Hausl, for their invitation to spend a couple of months in the Institute for Communications Engineering at the Technical University of Munich during the fall of 2007. During that period, I was very lucky to meet the late Prof. Ralf Koetter and collaborate with him. This led to a second visit to Munich in the summer of 2007 where I worked with

him and Prof. Michelle Effros. I wish to thank both of them for their hospitality and engaging and exciting research discussions. I would like also to acknowledge the help of my friend, Fakheredine Keyrouz, who always provides me accommodation when I am in Munich and makes my stay there very comfortable. My visits to Munich marked an important event in my life and that is meeting the two wisest men I have ever known: Fr. Dany Younes and Dr. William Bellis. I am deeply grateful to them for teaching me how to stay focused on what is important in life.

During my five years at Texas A&M, I shared my office with several friends: Dana Jaber, Fan Zhang and Daehyun Choi. I would like to thank them all for making our office a fun and stimulating workplace. Special thanks also to my friend, Mustapha El-Halabi, for his exquisite culinary skills and for his help in proofreading parts of this document. I am also grateful to many other friends who made living in College Station much more enjoyable. Among them I list Mary Abou Nader, Sujan Dan, Babak Fariabi, Chadi Geha, Jing Jiang, Haejun Kim, Andriy Nemchenko, Parimal Parag, Pheba Thomas, David and Debbie Rivera, and Golnaz Vahedi. I ask for forgiveness for those who were not mentioned for their real place is in my heart.

Last but not least, I owe my deepest gratitude to my parents. Without their sacrifice and unconditional love, this work would have not been possible.

TABLE OF CONTENTS

CHAPTER		Page
I	INTRODUCTION	1
	A. Communications Over Networks	1
	B. Routing Vs. Network Coding	2
	C. Background on Index Coding	6
	D. Overview and Contributions	9
	1. Network Coding for Network Robustness	10
	2. Secure Network Coding	14
	3. Index Codes and Matroid Representation	18
	E. Literature Overview	23
	F. Dissertation Outline	26
II	ROBUST NETWORK CODES FOR UNICAST CONNECTIONS	28
	A. Introduction	28
	B. Model and Preliminaries	34
	1. Network Codes	34
	2. Flow and Cut Conditions	35
	C. Minimal and Simple Networks	36
	1. Reduced Capacity Function	36
	2. Simple Networks	39
	D. Structure of Simple Networks	43
	1. Node Properties of Simple Unicast Networks	43
	2. Residual Graphs and Residual Cycles	44
	3. Block Decomposition	46
	4. Proof of Lemma D.4	49
	E. Network Codes for Simple Networks	53
	F. Minimizing the Required Amount of Network Resources . .	58
III	SECURE NETWORK CODING FOR WIRETAP NETWORKS OF TYPE II	61
	A. Introduction	61
	B. Wiretap Channel II	64
	C. Wiretap Network II	65
	D. Network Code Design and Alphabet Size	72

CHAPTER	Page
E. Wiretapper Equivocation	76
1. Wiretap Channel of Type II	78
2. Underestimated Wiretapper	79
3. Restricted Wiretapper	79
F. Connections with Other Schemes	81
1. Secure Network Coding and Filtered Secret Sharing	81
2. Universal Secure Network Codes	83
3. Byzantine Adversaries	84
IV NETWORK CODING, INDEX CODING AND MATROID THEORY	86
A. Introduction	86
B. Model	90
1. Index Coding	90
2. Network Coding	92
C. Connection to Network Coding	93
D. Connection to Matroid Theory	98
1. Overview of Matroid Theory	98
2. From Matroids to Index Codes	102
E. Properties of Index Codes	104
1. Scalar Vs. Vector Linear Codes	104
2. Linear Vs. Non-Linear Codes	106
F. From Matroids to Networks	107
V CONCLUSION	114
REFERENCES	117
APPENDIX A	128
VITA	132

LIST OF TABLES

TABLE	Page
I Packets received by the destination of the network of Figure 5(a) for the different single edge failure possibilities. In all cases, the destination receives two linearly independent combinations of x and y	13

LIST OF FIGURES

FIGURE	Page
1	(a) A unicast network with a single source n_1 and a single destination n_7 . (b) A maximal flow of value 2 from the source to the destination represented by two edge-disjoint paths depicted with dashed edges. 2
2	(a) The butterfly network is an example of a multicast network with a single source node n_1 and two destinations n_6 and n_7 . There are two information sources, x and y , that need to be delivered to both destinations. (b) A routing scheme: x is routed to both destinations along the tree formed by the dashed edges, y is only sent to n_7 . It is impossible to simultaneously satisfy the demands of both destinations using a routing scheme. 4
3	A network code for the butterfly network [1]. Each destination receives two linearly independent combinations, and thus can decode x and y 5
4	An instance of the index coding problem with four messages x_1, \dots, x_4 at the transmitter and four receivers each demanding one of the messages and has some side information. 8
5	(a) A unicast network with two sources x and y . All edges have unit capacities except edges $(1, 2)$ and $(1, 3)$ which are of capacity 2. The labels on the edges describe a proposed robust network code.(b) The network behavior when edge $(1, 2)$ fails. A failed edge can only forward all-zeros packets. Nevertheless, the destination can still decode. 11
6	(a) Basic building blocks for a unicast network. (b) Block decomposition of a simple unicast network [2, 3]. 12

FIGURE	Page
7	The wiretapped butterfly network with a linear multicast network code over \mathbb{F}_3 . Security is achieved against a wiretapper that can access one edge by using a coset code of parity check matrix $\mathcal{H} = \begin{bmatrix} 1 & 1 \end{bmatrix}$ on top of the network. 16
8	(a) The M-Network [4] with four packets x_1, x_2, x_3, x_4 at the source nodes and four destination nodes with non-multicast demands. (b) A vector linear network code of dimension 2 for the M-network. 18
9	The network constructed in [5] as a counterexample to the conjecture on the sufficiency of linear network codes. This network does not have a vector linear network code over any field, but has a non-linear one over a quaternary alphabet. 20
10	A network that is equivalent to the index coding instance of Figure 4. Dashed edges represent the side information available to the receivers in the index coding problem. 21
11	A network based on the non-Pappus matroid with three source nodes carrying three distinct information packets x_1, x_2 and x_3 . This network does not admit a scalar linear network code but a vector linear code of dimension 2 over \mathbb{F}_3 . The network comprises additional destinations that are not represented here for sake of clarity (see Chapter IV). 22
12	(a) Dedicated path protection method (1 + 1 path protection); (b) Diversity coding method for $h = 2$. The network edges are labeled by their capacities. 30
13	A network coding approach for $h = 2$ 32
14	(a) Node v of degree eight; (b) The intermediate step in constructing the gadget Γ_v ; (c) The final step in constructing the gadget Γ_v 40
15	The four possible types of nodes in a simple unicast network. 43

FIGURE	Page
16	(a) A graph $G(V, E)$ with edges of unit capacity and a flow θ of value three. Each edge $e \in E$ is labeled with the amount of flow $\theta(e)$ it carries. $\hat{E} = \{(v_1, v_4), (v_1, v_5), (v_2, v_4), (v_2, v_6), (v_3, v_5), (v_3, v_6)\}$. (b) Residual graph for $E_1 = \{(v_1, v_5), (v_2, v_4), (v_3, v_6)\}$. The graph contains a residual cycle $W = \{v_1, v_5, v_3, v_6, v_2, v_4, v_1\}$. (c) The flow θ' obtained from θ by augmenting along cycle W . Note that edges (v_1, v_5) , (v_2, v_4) , and (v_3, v_6) are redundant and can be removed from the network without violating its feasibility. 46
17	(a) The three basic building blocks of types A, B and C, for simple unicast networks. (b) An example of the block decomposition of a simple unicast network. 47
18	(a) and (b) Examples of cuts of Type 1; (c) and (d) Examples of cuts of Type 2. 48
19	(a) An example of a Type 1 cut with two nodes of Type I and one node of Type IV. (b) The corresponding graph G' with $E_1 = \{(u_2, x), (u_3, y)\}$ 51
20	Robust network code for simple unicast networks: (a) Encoding for blocks of Type A; (b) Encoding for blocks of Type B; (c) Encoding for blocks of Type C. 54
21	Network equivalent to the wiretap channel of type II. 66
22	Single-edge wiretap butterfly network with a) non-secure network code and b) secure network code. Security is achieved by using a coset encoder on top of the network. 67
23	The combination network $B(n, M)$ 70
24	A secure network code for the $B(3, 4)$ combination network based on a $[6, 3]$ Reed-Solomon code over \mathbb{F}_7 71
25	A coding scheme achieving perfect secrecy against a limited Byzantine wiretapper. 85

FIGURE	Page
26	An instance of the index coding problem with four messages and four receivers ρ_1, \dots, ρ_2 . Each receiver ρ_i is represented by a couple (x, H) , where $x \in X$ is the packet demanded by the receiver, and $H \subseteq X$ represent its side information. 88
27	An instance of the network coding problem equivalent to the instance of the index coding problem depicted in Figure 26. 94
28	A geometrical representation of the non-Pappus matroid [6, p.43]. The matroid circuits are represented by straight lines. 100
29	The M-Network \mathcal{N}_1 introduced in [4]. 105
30	The network \mathcal{N}_2 of [5]. \mathcal{N}_2 does not admit any vector linear network code, but has a non-linear one over a quaternary alphabet. . . 107
31	Part of the network resulting equivalent to the non-Pappus matroid resulting from the construction of Definition F.1. 110
32	A subnetwork of the network \mathcal{N}_3 111
33	A cut of Type 2. Each edge is labeled by the corresponding flow value. 130
34	Examples of subgraphs of non-minimal unicast graph that include a cut of Type 2. The labels on the edges represent the amount of flow they carry. Edge in E_1 are depicted by dashed lines. (a) An example of the case when all the nodes adjacent to u_1 and u_2 are distinct. (b) The corresponding residual graph with residual cycle $W = \{u_1, u_1^2, u_2^2, u_1^1, u_1\}$. (c) An example of the case when u_1^1 coincides with u_2^2 , but u_1^2 and u_2^1 are distinct nodes. The residual cycle in this case is $W = \{u_2, u_1^1, u_1, u_1^2, v, u_2^1, u_2\}$ 131

CHAPTER I

INTRODUCTION

A. Communications Over Networks

We live in an information age where easy access to information, anywhere and anytime, is no more a privilege, but rather a necessity for daily life. The huge commercial success of wireless systems in the recent decades, such as cellular and WiFi networks, has played a major role in making information accessibility even easier and more ubiquitous.

This progress is accompanied, in what seems to be a common trend in many sectors of information technology, by an ever-growing demand for bandwidth by popular services and applications on the user side. To satisfy this increasing demand, service providers are forced to invest in costly resources, such as base stations, optical fibers and cables, to be added to their networks. Under such circumstances, it is very crucial, before upgrading a network, to make sure that it is operating in an optimal way, and that the existing infrastructure is fully exploited.

From a theoretical point of view, studying the optimal operation of communication networks, which are typically characterized by noise and interference, can be a very difficult task, even for very small and simple networks such as the relay channel whose capacity is still unknown [7]. Even for network models where all the communication channels are assumed to be perfect and free of noise and interference, the problem of information transmission is far from being well understood, and many related interesting and challenging questions remain unanswered.

This dissertation studies wired and wireless communication networks under the

The journal model is *IEEE Transactions on Automatic Control*.

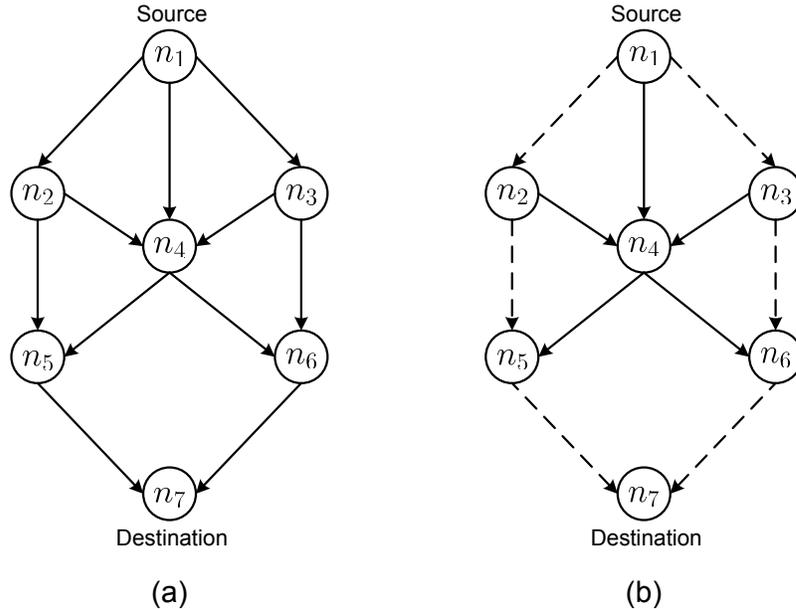


Fig. 1. (a) A unicast network with a single source n_1 and a single destination n_7 . (b) A maximal flow of value 2 from the source to the destination represented by two edge-disjoint paths depicted with dashed edges.

noise-free and interference-free model. Different mathematical aspects of the use of the novel techniques of network coding are investigated for guarantying a flow of information that is reliable, secure, and cost-effective.

B. Routing Vs. Network Coding

Consider the communication network represented by the directed graph depicted in Figure 1(a). The graph vertices, or nodes, represent transceivers such as servers, routers, mobile phones, TV sets, etc. The graph edges, or links, represent communication channels such as optical fibers, telephone lines, DSL cables, radio channels, etc. Those channels are assumed to be perfect and of unit capacity. The signal received at the head node of an edge is an exact replica of the one transmitted at its tail node. This network is called a *unicast* network since it has a single source node

n_1 , and a single destination node n_7 . The following question naturally arises: what is the maximum information rate, or maximum flow, that can be sent from the source to the destination, and how can it be achieved?

The answer to this question is given by the famous max-flow min-cut theorem [8] which states that, in a unicast network, the maximum value of a flow from the source to the destination is equal to the minimum capacity of a cut in the underlying graph¹. Moreover, there are efficient algorithms, such as the Ford-Fulkerson algorithm, that can find a maximum flow in a graph in polynomial time. In unicast networks with unit capacity edges, the value h of a maximum flow is always an integer and can be achieved by finding h edge-disjoint paths starting at the source and ending at the destination, and routing the information along those paths. The maximum value of a flow in the network of Figure 1(a) is equal to 2, and can be achieved by routing the information along the two edge-disjoint paths: (n_1, n_2, n_5, n_7) and (n_1, n_3, n_6, n_7) , depicted in Figure 1(b) with dashed edges, each carrying a unit rate information stream.

The problem becomes more challenging with networks with multiple sources and multiple destinations. *Multicast networks* form an important class of networks having a single source node and multiple destinations each demanding all the information available at the source. Figure 2 depicts a famous example of a multicast network, known as the *butterfly network*, where there are two information sources x and y at node n_1 , and two destinations, n_6 and n_7 , that both demand x and y . The information sources x and y are assumed to produce streams of incompressible bits at a rate of 1 MB/s, the links are also assumed to be perfect channels of capacity 1 MB/s. The

¹The concepts of flows and cuts in graphs will be defined rigorously in the next chapter.

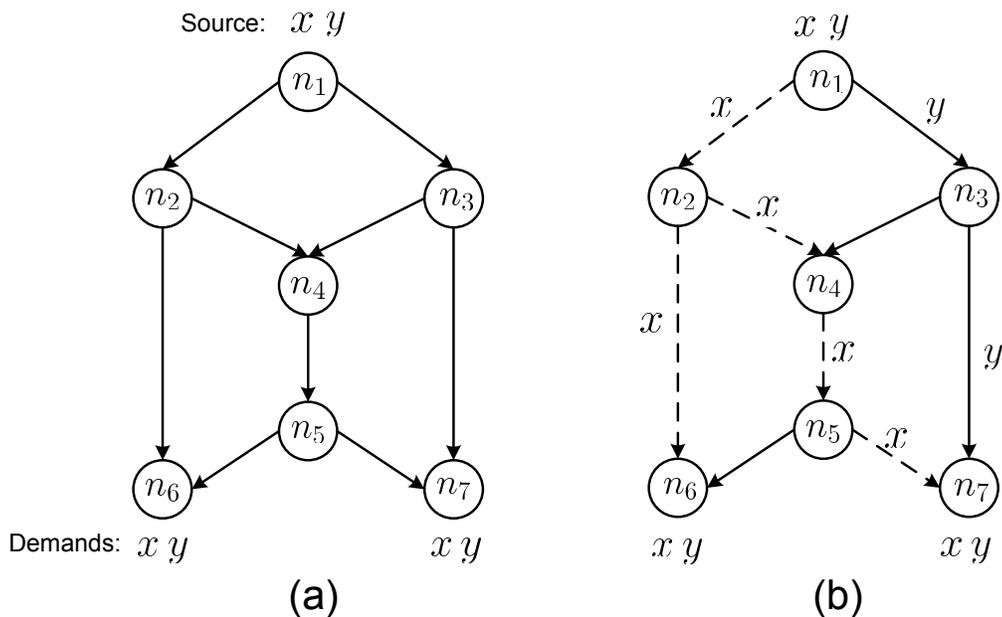


Fig. 2. (a) The butterfly network is an example of a multicast network with a single source node n_1 and two destinations n_6 and n_7 . There are two information sources, x and y , that need to be delivered to both destinations. (b) A routing scheme: x is routed to both destinations along the tree formed by the dashed edges, y is only sent to n_7 . It is impossible to simultaneously satisfy the demands of both destinations using a routing scheme.

traditional approach for the multicast case consists of looking at it as an instance of the *packing Steiner trees* problem, which is known to be an NP-hard [9] problem. A corresponding solution relies on finding two edge-independent trees in the graph (called Steiner trees), each having the source node as root, and the destination nodes as leaves. Each tree can then be used to forward the data along its edges from the source node to the destinations. One can check that any two different Steiner trees in the butterfly network would necessarily have to share an edge. Therefore, the demands of the two destinations cannot be satisfied simultaneously, and one can send only information at a rate of 1 MB/s from n_1 to both n_6 and n_7 along a single Steiner

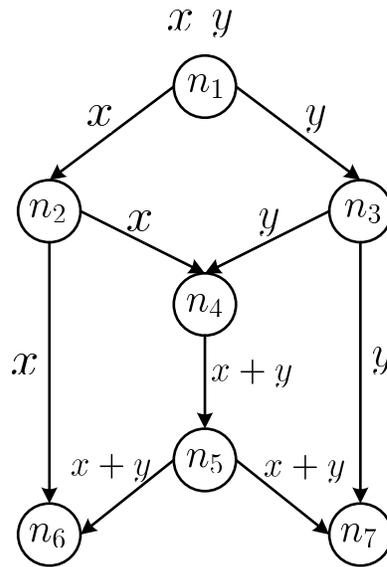


Fig. 3. A network code for the butterfly network [1]. Each destination receives two linearly independent combinations, and thus can decode x and y .

tree. For instance, x can be sent to both destination along the tree depicted in Figure 2(b) [1] with dashed edges. The redundant edges in the network can be used to send source y only to n_7 . A symmetric solution can deliver x and y to n_7 and x to n_6 . By time sharing between these two solutions, one can send a maximum average rate of 1.5 MB/s to both destinations. But, can we do better?

In the above approach, the operation of each node was restricted to *routing*, i.e. *copying and forwarding* the incoming data. This reflects the implicit assumption that information, similar to commodities, needs to be carried through the network in the same form that is available at the source. However, this is not really necessary, and the only requirement should be that all the destinations must be able to reconstruct their demands from what the information received on their incoming edges. Specifically, each node can be allowed to send on each of its outgoing edges any function of its incoming data as long as the above requirement is met.

By allowing this generalization, one can find a scheme that can satisfy the demands of the two destinations each at a rate of 2 MB/s. Such scheme is depicted in Figure 3 where each edge is labeled by the information it carries. The novelty of this scheme is that the output of node n_4 is neither x nor y , but $x + y$, where “+” denotes the Xor operation. Destination n_6 receives x and $x + y$ and thus can decode y by subtracting the two received packets ($y=(x+y)-x$). Similarly, destination n_7 can solve for x and y .

In this solution, node n_4 is said to be performing *network coding*. Network coding is the generalization of the operation of network nodes, beyond routing, to outputting any function of their incoming information, making it possible, as in the example above, to increase the throughput in networks. This simple and original idea was first introduced by Ahlswede *et al.* in their seminal paper [1] in year 2000 and has generated since then a huge interest in the research community.

C. Background on Index Coding

The example of the butterfly network demonstrates the benefits of encoding the data which can result in an increase of throughput in wired networks, even in noise-free and interference-free settings. In this section, we focus on wireless scenarios, and show that coding can be very advantageous in this case too, and may lead to many benefits such as energy savings and delay reduction.

The wireless medium has the particular characteristic of allowing a transmitter to deliver data to several neighboring receivers with a single transmission. Moreover, a wireless receiver can opportunistically listen to the wireless channel and store the overheard data, including those designated for other destinations. As a result, it can obtain side information which, in combination with proper encoding techniques, can

lead to a substantial improvement in the performance of the overall wireless network.

Consider for example a wireless scenario consisting of a single transmitter and a number of receivers. The transmitter has a set of information messages, say bits, $X = \{x_1, \dots, x_k\}$, that need to be delivered to the receivers. Each receiver demands a single message $x_i \in X$, and has a prior knowledge of a subset of X that he may have previously overheard. The transmitter can transmit information to the receivers through a noiseless broadcast channel having a capacity of one message per channel use. That is, in each transmission, the transmitter can broadcast a single message to all the receivers who will get it with no errors. Moreover, we assume that the transmitter knows the side information of each receiver, and that the receivers cannot communicate or cooperate among each others. Our objective is to find an optimal encoding scheme that satisfies all the receivers demands with minimum number of transmissions, i.e., with minimum uses of the broadcast channel.

An example of the scenario described above is depicted in Figure 4 which includes a transmitter with four messages $x_1, \dots, x_4 \in \{0, 1\}$, and four receivers with different demands and side information sets. Evidently, the transmitter can satisfy the demands of all the receivers, in a straightforward manner, by broadcasting all four messages over the wireless channel. This solution requires four transmissions. However, this number can be reduced by half by encoding the information at the transmitter. Indeed, it is sufficient to send just the two messages $x_1 + x_2 + x_3$ and $x_1 + x_4$ (addition is over \mathbb{F}_2) to satisfy the receivers demands. This example demonstrates that by using an efficient encoding scheme, the sender can significantly reduce the number of transmissions, which reduces the system delay and the energy consumption.

The source coding problem described above is referred to as the *index coding*

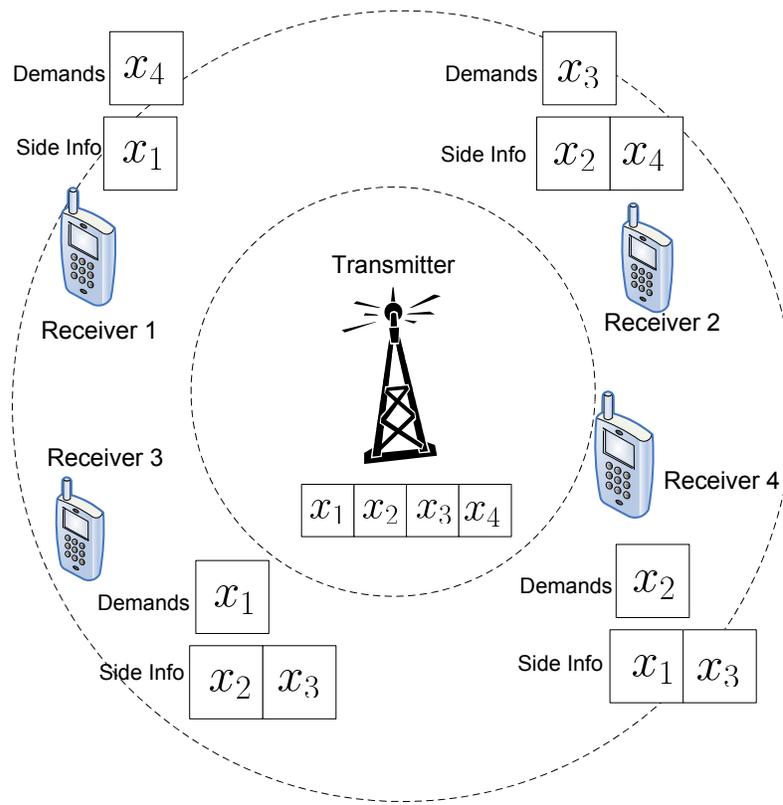


Fig. 4. An instance of the index coding problem with four messages x_1, \dots, x_4 at the transmitter and four receivers each demanding one of the messages and has some side information.

problem [10] in the literature, and may arise in different practical communication scenarios. For instance, Birk and Kol who first studied this problem in [11] were motivated by a satellite communication problem where a server broadcasts data packets to a number of caching clients in an attempt to reduce latency in the network, and balance the communication load. In a first stage, the server starts by transmitting the data to all the clients who store the received packets in their caches. Typically, each client is interested in a different set of data, and will only be able to store a subset of the transmitted data due to packets lost during transmission or even insufficient

storage capacity. In the second stage, the clients notify the server of their cache content via a low rate feedback channel. The server then initiates a second transmission session to fill the gaps in the client caches.

Index coding can also be useful in many other practical applications. For example, consider a peer-to-peer content distribution network that needs to deliver a set of large multimedia files to a number of clients. In this setting, if some parts of the files are already available to some clients, the distribution can be efficiently implemented by multicasting encoded chunks of the original files.

The previous problem leads to interesting questions pertaining to index codes: What is the minimum number of transmissions for a given instance? Is there an efficient algorithm to construct optimal or a near-optimal index codes? Are linear index codes always optimal? How do the alphabet size and other parameters affect the optimal solution? Some of these questions and other ones have already been answered in the literature [11, 10, 12]. In this dissertation, we present a new approach to understand this problem and answer some of the above questions by exploring its connection to network coding and matroid theory.

D. Overview and Contributions

The contributions of this dissertation can be organized in three categories. The first category consists of results on the design of robust network codes for protection against link failures in networks. The second category comprises results on the construction of network codes for achieving security in networks in the presence of an eavesdropper. The third category is focused on the index coding problem and its relation to network coding and matroid linear representation. Below, we summarize the results in each category.

1. Network Coding for Network Robustness

We consider the problem of establishing reliable unicast, i.e., single-source single-destination, connections across a communication network with non-uniform edge capacities. We assume that the edges in the network can fail, and thus, can no longer be useful for information transmission. Our objective is to provide communication schemes that are robust to such failures, and that achieve instantaneous recovery. The instantaneous recovery mechanisms ensure a continuous flow of the data from the source to the destination node with no interruption or data loss in the event of a failure. Such mechanisms eliminate the need for packet retransmission and rerouting. To that end, we study the construction of network codes that would achieve network robustness and guaranty instantaneous recovery in the event of a link failure as illustrated in the following example.

Figure 5(a) depicts a unicast network where, in each communication round, two packets x and y , belonging to some finite field, need to be sent from source node n_1 to the destination node n_5 . In this network, edges (n_1, n_2) and (n_1, n_3) can deliver two packets per communication round, whereas all the other edges have unit capacities. We assume that at most one link can fail at a time in the network, and that the output of the failed link is always the zero symbol. We propose the robust network code, depicted in the same figure which is capable of achieving instantaneous recovery from single edge failures. The crucial feature of this code is letting the intermediate node n_4 encode the packets received over its two incoming edges. Figure 5(b) depicts the behavior of this code when edge (n_1, n_2) fails. Despite the failure, the destination receives x on edge (n_4, n_5) and y on edge and (n_3, n_5) , and thus can instantaneously decode without notifying the source and retransmitting the information. This is actually true for any single edge failure, and Table I lists the messages received by

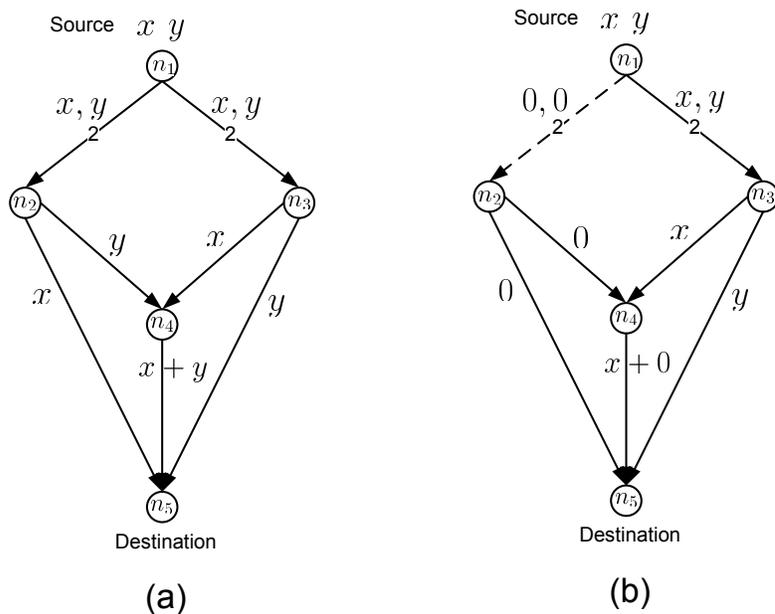


Fig. 5. (a) A unicast network with two sources x and y . All edges have unit capacities except edges $(1, 2)$ and $(1, 3)$ which are of capacity 2. The labels on the edges describe a proposed robust network code. (b) The network behavior when edge $(1, 2)$ fails. A failed edge can only forward all-zeros packets. Nevertheless, the destination can still decode.

the destination for the different failure possibilities. An underlying assumption here is that the transmitted packets always contain a header, typically of negligible length, indicating the encoding coefficients, which will be used by the destination to decode. It will be shown in Chapter II that instantaneous recovery is not possible without the encoding operation at the intermediate node n_4 . In particular, a scheme based on encoding the information at the source then routing it throughout the network cannot achieve instantaneous recovery.

We focus on the unicast case with two information sources, or equivalently, on the case where the stream of data at the source is allowed to be split into two sub-streams. This assumption is of practical importance because, in the typical operation

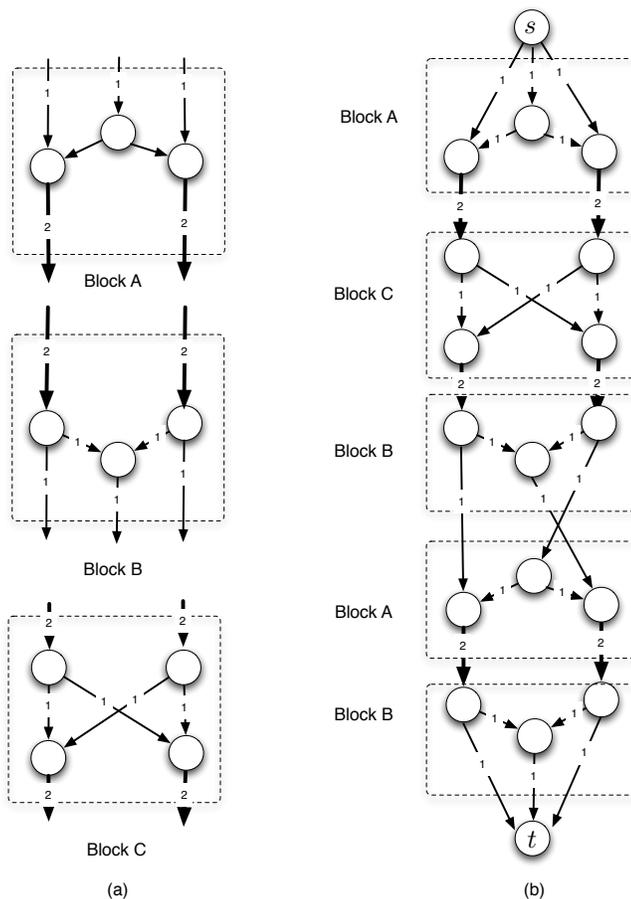


Fig. 6. (a) Basic building blocks for a unicast network. (b) Block decomposition of a simple unicast network [2, 3].

of real networks, it is unlikely that more than three disjoint paths will be allocated for a single connection.

First, we show that there is no loss of generality in restricting the study to a family of networks that we call *simple networks*, which are essentially characterized by the following two properties:

1. minimality: simple networks do not include redundant edges or edges of excessive capacity.

	(n_2, n_5)	(n_4, n_5)	(n_3, n_5)
no failure	x	$x + y$	y
(n_1, n_2)	x	y	0
(n_1, n_3)	0	x	y
(n_2, n_4)	x	x	y
(n_3, n_4)	x	y	y
(n_2, n_5)	0	$x + y$	y
(n_3, n_5)	x	0	y
(n_4, n_5)	x	$x + y$	0

Table I. Packets received by the destination of the network of Figure 5(a) for the different single edge failure possibilities. In all cases, the destination receives two linearly independent combinations of x and y .

2. uniform node degree: all the nodes in simple networks have a total degree of 3.

We demonstrate that simple networks have a unique combinatorial structure [2, 3]. More specifically, any simple network can be decomposed into the three basic building blocks of types A , B , and C depicted in Figure 6(a). Figure 6(b) depicts an example of such a network and its decomposition into five consecutive blocks of types A , C , B , A and B . Second, we use the block decomposition property of such networks to show the existence of robust network codes over the binary field \mathbb{F}_2 , and devise an algorithm that constructs such codes in an efficient manner [2, 3].

A robust network code for both unicast and multicast networks can be established through the standard network coding algorithm presented in [13]. However, this algorithm is designed for general failure patterns, and for our special single failure case, it requires a field size of $O(|E|)$, where E is the set of network edges. In

contrast, our scheme requires a small field size (\mathbb{F}_2) which does not depend on the number of edges in the underlying network. The size of the finite field is a very important factor in practical implementation schemes [14] as it determines the amount of communication and computational overhead. Our algorithm has a significantly smaller computational complexity associated with finding a robust network code than the existing solutions. Specifically, the computational complexity of our algorithm is $O(|V|^2)$, V is the number of nodes in the network, compared with $O(|E|^2)$ incurred by application of the algorithm due to [13].

We also address the problem of efficient allocation of network resources for a robust network. Again, we exploit the properties of minimal networks to devise another algorithm for finding a feasible solution whose cost is at most two times more than the optimum [2, 3]. To the best of our knowledge, this is the best approximation ratio for the problem at hand reported in the literature.

2. Secure Network Coding

We consider here networks that are susceptible to another type of vulnerability, which is wiretapping. We focus on networks with multicast demands in the presence of a wiretapper that can access data on a limited number of edges of his choice and where there are no shared randomness (“keys”) between the source and the destination nodes. Our primary goal is to design efficient network coding schemes that deliver data at a maximum rate to all the destinations and does not reveal any information about the transmitted data to the wiretapper.

The problem of making a linear network code information-theoretically secure in the presence of a wiretapper that can look at a bounded number of network edges was first studied by Cai and Yeung in [15]. In a network where the min-cut value between the source and each destination is n , and an adversary can access up to μ edges of

his choice, they constructed codes over an alphabet with at least $\binom{|E|}{\mu}$ elements which can support a secure multicast rate of up to $n - \mu$. The algorithm due to [15] has high computational complexity and requires a very large field that is exponential in the number of wiretapped edges.

We propose an alternative approach [16, 17] to this problem by regarding it as a network generalization of the Ozarow-Wyner wiretap channel of type II introduced in [18] and [19]. Our method consists of using a coding scheme at the source node that ensures information-theoretic security, and that is based on the Ozarow-Wyner coding scheme for the wiretap channel of type II, where the source transmits n symbols to the receiver and an adversary can access any μ of those symbols. Ozarow and Wyner showed that the maximum number k of symbols that the source can communicate securely through the wiretap channel of type II to the receiver is equal to $n - \mu$. They also proposed the *coset coding* scheme to achieve this secure transmission rate. Clearly, if the n channel symbols are multicast over a network using a routing scheme, the k source symbols remain secure in the presence of an adversary with access to any μ edges. We will illustrate in Chapter III that, however, this is not necessarily the case when network coding is used. We will show that a secure network code based on the Ozarow-Wyner scheme can still be designed over a sufficiently large field.

The coset code scheme proposed by Ozarow and Wyner is defined by an $[n, n - k]$ linear block code $\mathcal{C} \subset \mathbb{F}_q^n$ with a $k \times n$ parity check matrix \mathcal{H} ($\mathcal{H}X^T = 0; \forall X \in \mathcal{C}$). In contrast to classical error-correcting codes which are deterministic in general, a coset code is a random encoding function that operates in the following way: to send a message $S = (s_1, \dots, s_k) \in \mathbb{F}_q^k$, the coset code outputs a random element $Y = (y_1, \dots, y_n) \in \mathbb{F}_q^n$ picked uniformly from the coset $S + \mathcal{C}$ of the code \mathcal{C} , i.e., a random element of the coset that has S as syndrome. The output Y of the encoder

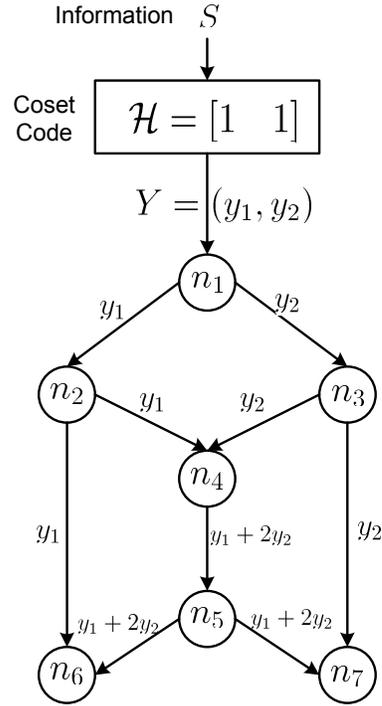


Fig. 7. The wiretapped butterfly network with a linear multicast network code over \mathbb{F}_3 . Security is achieved against a wiretapper that can access one edge by using a coset code of parity check matrix $\mathcal{H} = [1 \ 1]$ on top of the network.

is thus a randomly picked solution of the under-determined linear system $\mathcal{H}Y = S$. Figure 7 shows how we propose to use this scheme for multicast networks by encoding the information using a coset code at the source before injecting the output in the network. The figure shows the butterfly network with min-cut $n = 2$ and a corresponding linear multicast network code over \mathbb{F}_3 in the presence of a wiretapper that can access a single edge of his choice ($\mu = 1$). At most one symbol $S \in \mathbb{F}_3$ can be sent securely in this case, and this can be achieved by using a coset code with parity check matrix $\mathcal{H} = [1 \ 1]$ on top of the network. In other words, the output $Y = (y_1, y_2)$ of the coset code is a random solution of the equation $y_1 + y_2 = S$ over \mathbb{F}_3 . It can be seen that the wiretapper does not gain any information by observing

the messages on any single link. For instance, suppose that the wiretapper observes that the message on edge (n_4, n_5) is 0, i.e., $y_1 + 2y_2 = 0$. Then, he would know that there are three possible values of (y_1, y_3) , namely $(0, 0)$, $(1, 1)$ and $(2, 2)$, and therefore, the corresponding possible values of S , based on the coset code equation $S = y_1 + y_2$, are respectively 0, 2 and 1 which can occur with equal probabilities. Therefore, the wiretapper is still as uncertain about the value of S as prior to making his observation. An important issue in the design of a coset code is the choice of the parity check matrix \mathcal{H} , which is crucial in achieving security. For instance, using a coset code with $\mathcal{H} = [1 \ 2]$ will breach the security constraint since the wiretapper would be able to know exactly the value of S by tapping into any of the three edges (n_4, n_5) , (n_5, n_6) or (n_5, n_7) .

Given a multicast network with a linear network code, we derive the necessary and sufficient conditions on the parity check matrix \mathcal{H} of an Ozarow-Wyner coset code to guaranty perfect secrecy and prevent revealing any information to the wiretapper [16, 17]. We show that our scheme is equivalent to the one proposed in the work of Cai and Yeung in [15]. However, with our approach, we can quickly and transparently recover many of the results available in the literature on secure network coding. Furthermore, we use the results on the encoding complexity of the network coding schemes [20], [21] to derive new bounds on the required field size for a secure linear network code that are independent of the number of edges in the network, and that depend only on the number k of source symbols and the number of destinations. We also propose an algorithm for the construction of secure network codes that achieve these bounds. Furthermore, we look at the dual problem and analyze the security of a given coset code by studying the amount of information that is leaked to the wiretapper as a function of the number of wiretapped edges.

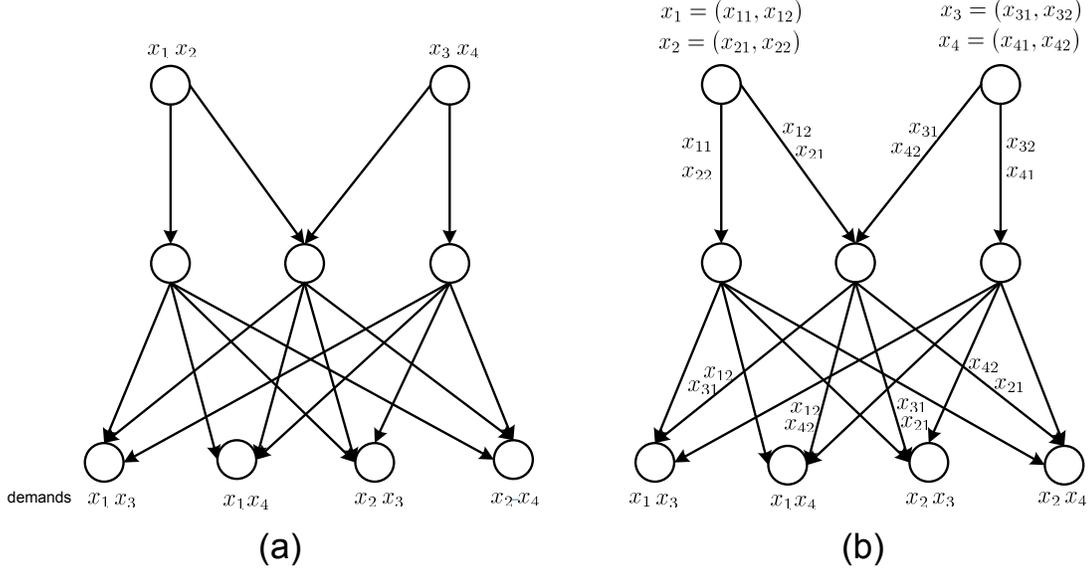


Fig. 8. (a) The M-Network [4] with four packets x_1, x_2, x_3, x_4 at the source nodes and four destination nodes with non-multicast demands. (b) A vector linear network code of dimension 2 for the M-network.

3. Index Codes and Matroid Representation

The network coding literature distinguishes between several categories of network codes based on the type of the edge encoding functions. The first natural distinction is between linear and non-linear codes. Linear codes have been extensively studied due to their tractability and importance for practical implementations, and, in their turn, they are divided into two subclasses: *scalar linear* and *vector linear*. In a vector linear network code, the original information packets available at the source nodes are modeled as vectors of dimension n with coordinates in \mathbb{F}_q , and therefore can be regarded as elements of \mathbb{F}_q^n . In addition, the edge encoding operations are linear functions of the packets coordinates over the base field \mathbb{F}_q . Scalar linear network codes are vector linear codes of dimension one $n = 1$. The network coding examples given

previously, for instance the one in Figure 3, are all scalar linear. A network admitting a scalar linear network code will consequently have a vector linear network code of any dimension over the same field. The converse, however, is not true. Médard *et al.* presented in [4] a counterexample consisting of the M-network depicted in Figure 8(a) which has the interesting property that it does not admit a scalar linear network code but it has a vector linear one. Figure 8(b) depicts a vector linear network code of dimension 2 for the M-network which corresponds to a simple routing scheme. The dimension n of vector linear codes can have two interpretations. From an information-theoretic perspective, it can be regarded as equivalent to the block length of linear error correcting codes, where, in this case, the network is used n -times. From a networking perspective, vector linear codes can be looked at as a fractional solution where each packet at the source is divided into n sub-packets.

Médard *et al.* conjectured in [4] that linear network codes, in their vector form, are sufficient to achieve the capacity of general networks. This conjecture was later disproved by Dougherty *et al.* in [5] where the network of Figure 9 was constructed. It was shown that this network does not admit any vector linear network code, but has a non-linear vector code over an alphabet of size 4. The authors based their construction on the linear representation properties of the Fano and non-Fano matroids to build two networks, one admitting vector linear network codes over fields with even characteristic, and the other over fields with odd characteristic. The counterexample network of Figure 9 is then obtained by juxtaposing these two networks.

Index codes can be subject to the same taxonomy. One can distinguish among non-linear, scalar linear and vector linear index codes. A question that naturally arises here is whether scalar or vector linear index codes are optimal, i.e., whether the minimum number of transmissions can be always achieved by linearly encoding the

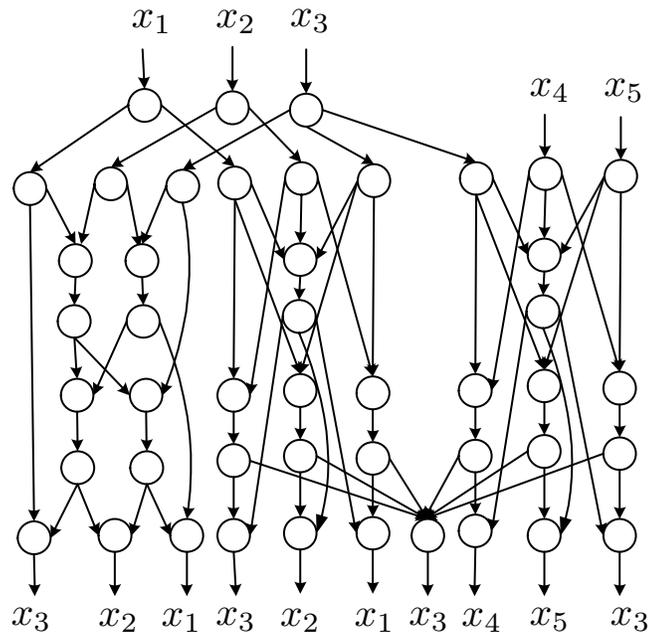


Fig. 9. The network constructed in [5] as a counterexample to the conjecture on the sufficiency of linear network codes. This network does not have a vector linear network code over any field, but has a non-linear one over a quaternary alphabet.

information, in a scalar or vector manner, at the transmitter. Motivated by a number of instances where scalar linear index codes over the binary field were optimal, Bar-Yossef *et al.* conjectured in [10] that these codes are optimal in general. Lubetzky and Stav disproved this conjecture in [12] and constructed a family of counterexamples where non-linear index codes are optimum and scalar linear codes are not. They concluded their paper by asking whether vector linear codes can outperform scalar linear codes. This question was answered positively later by Alon *et al.* in [22].

In this dissertation, we concurrently answer the questions on the optimality of scalar and vector linear index codes by exploring the connection between index coding and network coding. Index coding can be thought of as a special case of network

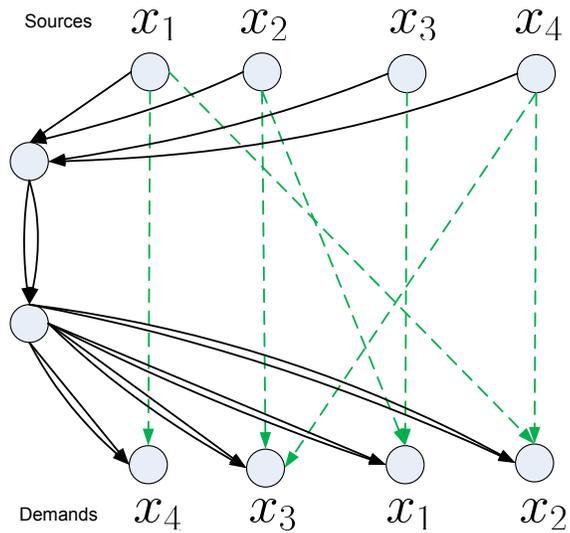


Fig. 10. A network that is equivalent to the index coding instance of Figure 4. Dashed edges represent the side information available to the receivers in the index coding problem.

coding. For instance, the index coding scenario of Figure 4 is equivalent to the network in Figure 10 where the information packets at the transmitter correspond to source nodes in the network, the receivers to destination nodes, and the side information to direct edges from the source to the destinations. In this case, it can be seen that a network code exists for this network if and only if there exists an index code consisting of two transmissions.

We also demonstrate in Chapter IV that, when restricted to linear codes, the index coding problem is equivalent to the more general network coding problem. To that end, we establish a reduction that maps any instance of the network coding problem to a corresponding instance of the index coding problem such that a vector linear network code for the given network implies an optimal vector linear index code of the same dimension over the same field and vice versa [23, 24, 25]. As a result,

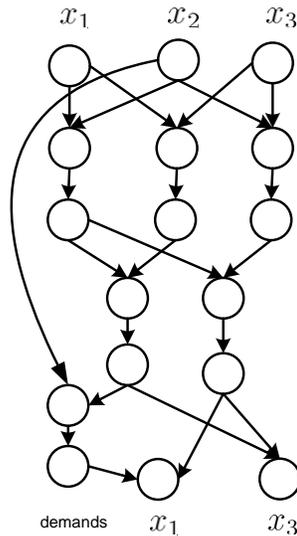


Fig. 11. A network based on the non-Pappus matroid with three source nodes carrying three distinct information packets x_1, x_2 and x_3 . This network does not admit a scalar linear network code but a vector linear code of dimension 2 over \mathbb{F}_3 . The network comprises additional destinations that are not represented here for sake of clarity (see Chapter IV).

several important properties of the network coding problem can be carried over to index coding. Specifically, by applying our reduction to the network of [5] depicted in Figure 9, we construct another counterexample to the conjecture in [10] where, not only binary scalar linear index codes are suboptimal, but also vector linear index codes are as well. Moreover, using this reduction in conjunction with the properties of the M-network, we construct an instance of the index coding problem in which vector linear index codes yield a smaller number of transmissions than scalar linear ones.

We also follow another direction in studying index codes by investigating their relation to matroid representation. We present a second reduction that maps any given matroid to an index coding problem where optimal vector linear index codes

corresponds to the matroid linear representation, and vice versa [23, 24, 25]. This construction also establishes a strong relation between network coding and matroid theory, and constitutes a means to apply numerous results in the rich field of matroid theory to communication problems in networks.

In contrast to the method described in [26], the network obtained by this construction has the property that any corresponding linear network code will directly induce a linear representation of the given matroid. This is due to the fact that the constructed network reflects all the dependency and independency relations of the given matroid. As an application of this reduction, we present the new network partially depicted in Figure 11, based on the non-Pappus matroid, that, similarly to the M-network, does not admit a scalar linear code but has a vector linear one.

E. Literature Overview

The network coding technique has been introduced in the seminal paper of Ahlswede *et al.* in [1]. Initial works on network coding focused on establishing *multicast* connections. It was shown in [1] and [27] that the capacity of a multicast network, i.e., the maximum number of packets that can be sent from the source node to a set of terminals per time unit, is equal to the minimum capacity of all the cuts that separates the source from any terminal. In a subsequent work, Koetter and Médard [28] developed an algebraic framework for network coding and investigated linear network codes for directed graphs. This framework was used by Ho *et al.* [29] to show that linear network codes can be efficiently constructed through a randomized algorithm. Network coding for networks with cycles has been also studied in [30] and [31]. Comprehensive surveys on network coding theory can be found in the books [32, 33, 34], and [35].

The idea of using network coding for instantaneous recovery from edge failures was first described by Koetter and Médard [28]. They showed that if the network has a sufficient capacity to recover (e.g., by rerouting) from different separate failure scenarios, then there exists a linear network code, referred to as *robust network code*, that can simultaneously protect against all these failures and achieve instantaneous recovery. Jaggi *et al.* in [13] presented a polynomial-time algorithm for finding robust linear network codes. A different model for protection against erasures and also errors in networks was introduced by Koetter and Kschischang in [36] where communication is established by transmitting subspaces instead of vectors through the network. In [37], an information-theoretic framework for network management for recovery from edge failures has been presented. Failure protection schemes based on network coding were devised in [38, 39] for overlay networks. Using network coding for reliable communication was also discussed in [40] and [41]. References [14] and [42] describe practical implementations of network coding, and demonstrate its benefits for improving the reliability and robustness of networks. The problem of minimizing the amount of network resources allocated in a network has been considered in [43].

The problem of making a linear network code information-theoretically secure in the presence of a wiretapper that can look at a bounded number of network edges was first studied by Cai and Yeung in [15]. They considered directed graphs and constructed codes that can achieve the network multicast secrecy capacity. In [44], they proved that these codes use the minimum amount of randomness required to achieve the security constraint. However, the algorithm due to [15] has high computational complexity and requires a very large field size (exponential in the number of wiretapped edges). Feldman *et al.* derived trade-offs between security, code alphabet size, and multicast rate of secure network coding schemes in [45], by using ideas from secret sharing and abstracting the network topology. Another approach was taken by

Jain in [46] who obtained security by merely exploiting the topology of the underlying network. Weakly secure network codes that insure that no meaningful information is revealed to the adversary were studied in [47, 48].

Secure network coding in the presence of a Byzantine adversary that can modify the packets on the edges it controls has been studied by Ho *et al.* in [49] and Jaggi *et al.* in [50, 51, 52]. The problem of error correction in networks was also studied by Cai and Yeung in [53, 54] where they generalized classical error-correction coding techniques to network settings. The use of rank-metric codes for error control under this model was investigated in [55]. Silva and Kschischang used some of the results presented in [16] to construct universal secure network codes based on maximum rank-distance (MRD) codes [56, 57], and by Mills *et al.* in [58] to achieve secrecy for wireless erasure networks.

The index coding problem has been introduced by Birk and Kol [10] and was initially motivated by broadcast satellite applications². They developed several heuristic solutions for this problem and proposed protocols for practical implementation in satellite networks. Bar-Yossef *et al.* studied the index coding problem from a graph-theoretical perspective [10] and showed that the number of transmissions of an optimal linear index coding problem can be expressed as a certain functional, referred to as *minrank*, of a certain graphs. Finding the minrank of a graph, however, was proved to be an intractable problem [59]. Lubetzky and Stav [12] showed that non-linear scalar codes have a significant advantage over linear ones by constructing a family of instances with an increasing gap between the optimal number of transmissions required by non-linear and linear codes. Alon *et al.* studied in [22] the asymptotic behavior of a number of parameters pertaining to the index coding problem and showed that vector

²Reference [10] refers to the index coding problem as Informed Source Coding on Demand problem (ISCOD).

linear codes can have a better performance than scalar ones. Wu *et al.* [60] studied the information-theoretic aspects of the problem with the goal of characterizing the admissible rate region³. Reference [61] analyzed the hardness of approximation of the index coding problem. References [62] and [63] presented several heuristic solutions based on graph coloring and SAT solvers.

Matroids were first introduced and studied by Whitney [64] in 1935 in an effort to capture the abstract properties of the notion of dependence encountered in several disciplines, such as linear algebra. References [6] and [65] can be consulted for a detailed exposition of this theory. Dougherty *et al.* [5, 26] investigated the application of matroid theory to the general problem of information flow in networks. They introduced the class of matroidal networks, and described a method for building a matroidal network from a given matroid. This construction has been applied to specific matroids to prove important results in the field such as the insufficiency of Shannon-type information inequalities and linear network coding for, respectively, computing and achieving network capacity. The authors of [66] also studied the relation between the structure of multicast networks and certain matroids.

F. Dissertation Outline

This dissertation is organized as follows. In Chapter II, we present our results on the construction of robust network codes for unicast networks. In Chapter III, we discuss our generalization of the wiretap channel of type II of Ozarow and Wyner to the network setting and describe our secure network code scheme based on coset coding. In Chapter IV, we focus on the index coding problem and its relation to network coding and matroid linear representation and show that vector linear index

³Reference [60] refers to the index coding problem as the “Local Mixing Problem”.

codes are not always optimal.

CHAPTER II

ROBUST NETWORK CODES FOR UNICAST CONNECTIONS

We consider the problem of establishing reliable unicast connections across a communication network with non-uniform edge capacities. Our goal is to provide *instantaneous recovery* from single edge failures. With instantaneous recovery, the destination node can decode the packets sent by the source even if one of the network edges fails, without the need of retransmission or rerouting.

It has been recognized that, for this problem, *network coding* offers significant advantages over standard solutions such as disjoint path routing and diversity coding. Focusing on a practically important case in which the sender needs to deliver two packets per communication round, we present an efficient network coding algorithm over a small finite field ($GF(2)$). The small size of the underlying field results in a significant reduction in the computational and communication overhead associated with the practical implementation of the network coding technique. Our algorithm exploits the unique structure of *minimum coding networks*, i.e., networks that do not contain redundant edges.

We also consider the related capacity reservation problem and present an algorithm that achieves an approximation ratio of two compared to the optimal solution. The results reported in this chapter have appeared in [2, 3].

A. Introduction

In recent years, a significant effort has been devoted to improving the resilience of communication networks to failures and increasing their survivability. Edge failures are frequent in communication networks due to the inherent vulnerability of the communication infrastructure [67]. With the dramatic increase in data transmission rates,

even a single failure may result in vast data losses and cause major service disruptions for many users. Accordingly, there is a significant interest in improving network recovery mechanisms that enable a continuous flow of data from the source to the destination with minimal damage in the event of a failure.

Edge failures may occur due to several reasons, such as physical damage, misconfiguration, or a human error. Networks are typically designed to be resilient against a single edge failure. Indeed, protection from multiple failures incurs high costs in terms of network utilization, which is usually not justified by the rare occurrence of such failures.

We consider here the problem of establishing reliable unicast (single-source single-destination) connections across a communication network with non-uniform edge capacities. Our goal is to provide *instantaneous recovery* from single edge failures. The instantaneous recovery mechanisms ensure a continuous flow of data from the source to the destination node, with no interruption or data loss in the event of a failure. Such mechanisms eliminate the need of packets retransmission and rerouting. Instantaneous recovery is typically achieved by sending packets over multiple paths in a way that ensures that the destination node can recover the data it needs from the received packets. Below, we discuss three major techniques for achieving instantaneous recovery: *dedicated path protection scheme*, *diversity coding*, and *network coding*.

Network model. We model the communication network as a directed graph $G(V, E)$. We assume that each packet is an element of a certain finite field $\mathbb{F} = GF(2^m)$, where m is the packet length (in bits). We also assume that the data exchange is performed in rounds, such that each edge $e \in E$ can transmit $c(e)$ packets per communication round. We assume that $c(e)$ is an integer number, and refer to it as the *capacity* of edge e . The goal of a unicast connection is to transmit data from the source node $s \in V$ to the destination node $t \in V$. The *rate* h of the unicast

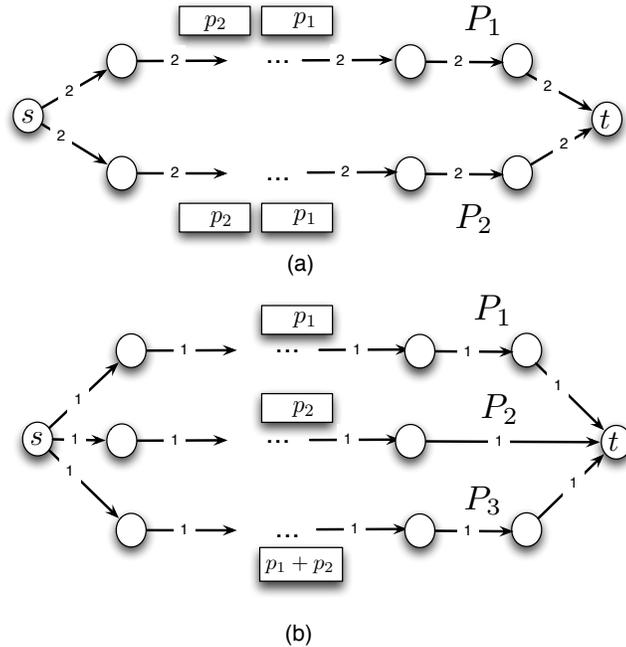


Fig. 12. (a) Dedicated path protection method (1 + 1 path protection); (b) Diversity coding method for $h = 2$. The network edges are labeled by their capacities.

connection is defined as the number of packets that are to be delivered from s to t per communication round. The capacity of an (s, t) -path P , from node s to node t , is defined to be the minimum capacity of an edge that belongs to P . In the normal state, each edge represents a lossless delay-free communication channel. However, network edges can fail, but at most one of the network edges can be faulty at any given time. We assume that a failed edge cannot transmit any data, and that an edge failure can be detected by its head node.

Dedicated path protection scheme. There are several techniques to achieve instantaneous recovery. A standard one employed in networks nowadays is the 1 + 1 *dedicated path protection scheme* [67]. This approach requires provisioning two disjoint paths P_1 and P_2 between s and t (see Figure 12(a)). Each packet generated

by the source node is sent over both paths, P_1 and P_2 . In the case of a single edge failure, at least one of the paths remains operational, hence the destination node will be able to receive the data without interruption. With this scheme both P_1 or P_2 must be of capacity at least h . While the dedicated path protection scheme is simple and easy to implement, it incurs high communication overhead due to the need to transmit two copies of each packet. In addition, it requires two disjoint paths which include edges of high capacity.

The diversity coding technique. The *diversity coding* technique [68] extends the dedicated path protection scheme by using multiple disjoint paths for sending the data. Figure 12(b) shows an example of a diversity coding scheme that uses three disjoint paths P_1, P_2 , and P_3 between s and t . The first two paths, P_1 and P_2 transmit the original packets, while P_3 transmits parity check packets. More specifically, for $h = 2$, paths P_1 and P_2 transmit packets p_1 and p_2 , respectively, while path P_3 transmits the packet $p_1 + p_2$, where p_1 and p_2 are the packets that need to be transmitted during the current communication round. Note that three disjoint paths can also be used for larger values of h by appropriate scaling of the edge capacities. Specifically, suppose that h is an even number and let $p_1, p_2, p_3, \dots, p_h$ be a set of packets that need to be transmitted during the current round. Then, path P_1 transmits odd packets p_1, p_3, \dots, p_{h-1} , path P_2 transmits even packets p_2, p_4, \dots, p_h , and path P_3 transmits parity check packets $p_1 + p_2, \dots, p_{h-1} + p_h$ (all operations are over $GF(2)$). Note that each path P_1, P_2 , and P_3 must have capacity of at least $\frac{h}{2}$. In general, the diversity coding scheme may include $k > 3$ disjoint paths. In this case, each of the paths transmits $\lceil \frac{h}{k-1} \rceil$ packets per round. Hence, the capacity of each of the paths must be at least $\lceil \frac{h}{k-1} \rceil$.

The network coding technique. While the disjoint coding technique offers

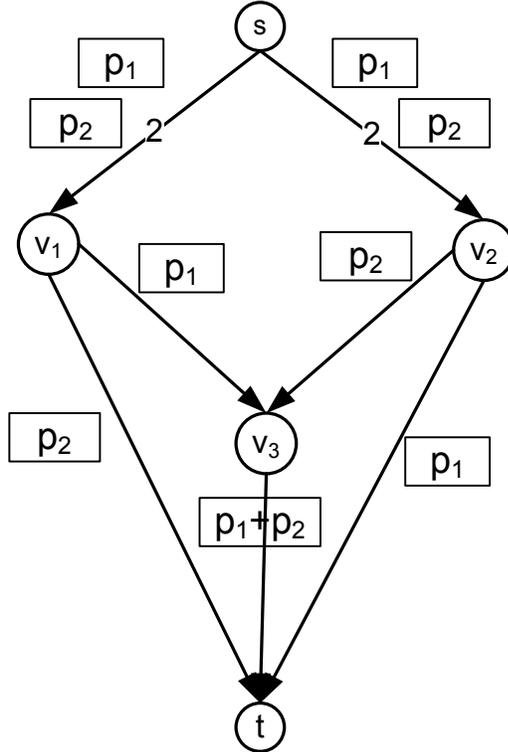


Fig. 13. A network coding approach for $h = 2$.

more flexibility than the dedicated path protection scheme, it is not the most general approach. For example, consider the network depicted in Figure 13. In this network, edges (s, v_1) and (s, v_2) have capacity two, while all other edges have unit capacity. Our goal is to establish a unicast connection that delivers two packets from s to t per communication round. We note that this network does not contain two disjoint paths of capacity two between s and t , hence the dedicated path protection scheme cannot be used. Furthermore, the diversity coding approach cannot be used as well because this network does not have three disjoint paths from s to t . However, instantaneous recovery from edge failures can be achieved by using the network coding approach. With this approach, the intermediate node v_3 combines packets received on its two incoming edges. The destination node can decode the packets sent by the

source node in any single edge failure scenario (see Table I). Note that without the encoding operation at the intermediate node v_3 , instantaneous recovery would not be possible. In fact, network coding is the most general approach for providing instantaneous recovery from edge failures. In particular, the network coding approach enables instantaneous recovery for any settings where such recovery is possible.

Path diversity vs. capacity requirements. Note that in diversity coding , there is a trade-off between the path diversity and the capacity requirement of the disjoint paths. In particular, the larger is the number of paths between the source and the destination nodes the lower is the capacity requirement on the paths, and, as a result, the smaller is the total communication overhead. Specifically, for the diversity coding scheme with k disjoint paths, the capacity requirement is equal to $\lceil \frac{h}{k-1} \rceil$, and the total amount of data sent over the network is equal to $k \cdot \lceil \frac{h}{k-1} \rceil$ per round. We observe that we can assume here, without loss of generality, that $h = k - 1$. Indeed, the larger values of h can be handled by scaling edge capacities.

A similar trade-off exists for the network coding approach. In this chapter, we restrict our attention to network topologies that correspond to $h = 2$. Intuitively, each network topology we consider can be divided into several parts, each part is either comprised of two disjoint paths of capacity two or three disjoint paths of capacity one. The network coding operations must be performed on some of the nodes that connect these parts. This has practical importance because, in a typical network scenario, it is unlikely that more than three disjoint paths will be used for a single connection. Note that our approach can be used for sending more than two packets per time unit by appropriate scaling of edge capacities.

B. Model and Preliminaries

1. Network Codes

For clarity of presentation, we define an auxiliary graph $\hat{G}(V, A)$ formed by the network graph $G(V, E)$ where each edge $e \in E$ is substituted by $c(e)$ parallel *arcs* that have the same tail and head nodes as e ; each arc can transmit one packet per round. We denote by $A(e) \subseteq A$ the set of arcs that correspond to edge e . In what follows we only refer to packets sent at the current communication round. The packets sent in the subsequent rounds are handled in a similar manner.

We denote by $\mathbb{P} = \{p_1, p_2, \dots, p_h\}$ the set containing the h packets that need to be delivered from s to t at the current communication round. A network code is defined by associating with each arc $a(v, u) \in A$ in the network an encoding function f_a that specifies the packet transmitted on arc a each time unit. For each arc $a(s, u) \in A(E)$, f_a is a function of the original h packets \mathbb{P} , i.e., $f_a : \mathbb{F}^h \rightarrow \mathbb{F}$. For each arc $a(v, u) \in A(E)$, $v \neq s$, f_a is a function of the packets received by node v at the current round, i.e., $f_a : \mathbb{F}^l \rightarrow \mathbb{F}$, where l is the number of incoming arcs of v in \hat{G} . A network code \mathbb{C} is a set of encoding functions associated with the arcs in $A(E)$, i.e., $\mathbb{C} = \{f_a \mid a \in A(E)\}$. In this chapter, we only consider *scalar linear network codes* where all packets are elements of a finite field and all encoding functions are also linear over that field. Our results show that there is no loss of generality incurring from this assumption.

As mentioned in the introduction, we assume that only one of the edges in the network can fail at a time. Since a failed edge e cannot transmit packets, we assume that the encoding function f_a of each arc $a \in A(e)$ is identically equal to zero, i.e., $f_a \equiv \mathbf{0}$. To guaranty instantaneous recovery, it is sufficient to ensure that for each edge failure there exists a set of h linearly independent packets among the packets

received by t .

Definition B.1 (Robust Network Codes). A network code \mathbb{C} is said to be *robust*, or *resilient to single edge failures*, if for each $e \in E$ it holds that the destination node t can decode the h packets sent by the source node s when all arcs in $A(e)$ fail.

2. Flow and Cut Conditions

A cut $C = (V_1, V_2)$ in the graph $G(V, E)$ is a partition of the nodes of V into two subsets V_1 and $V_2 = V \setminus V_1$. We say that a cut $C = (V_1, V_2)$ is an (s, t) -cut if it separates nodes s and t , i.e., if $s \in V_1$ and $t \in V_2$. We say that an edge $e \in E$ belongs to the cut (V_1, V_2) if its tail node belongs to V_1 , and its head node belongs to V_2 . The capacity of the cut is defined as the total capacity of all the edges that belong to the cut.

An (s, t) -flow θ in a graph $G(V, E)$ is a function $\theta : E \mapsto \mathbb{R}$ that satisfies the following two properties:

1. For all $e(u, v) \in E$, it holds that $0 \leq \theta(e) \leq c(e)$;
2. For each internal node $v \in V$, $v \neq s$, $v \notin t$, it holds that

$$\sum_{(w,v) \in E} \theta((w, v)) = \sum_{(v,w) \in E} \theta((v, w)).$$

The *value* $|\theta|$ of a flow θ is defined as

$$|\theta| = \sum_{(s,v) \in E} \theta((s, v)) - \sum_{(v,s) \in E} \theta((v, s)).$$

The cost $\omega(\theta)$ of a flow θ is defined as

$$\omega(\theta) = \sum_{(u,v) \in E} \theta((u, v)) \cdot m_e,$$

where m_e is the cost of reserving a unit capacity on edge e . Throughout this chapter, except for Section F, we assume that $m_e = 1$ for all $e \in E$.

A necessary condition for instantaneous recovery is that, for each $e \in E$, a network G^e formed from G by removing e must admit an (s, t) -flow of value h . By the max-flow min-cut theorem [69] this condition is equivalent to

$$\min_C \left[\sum_{e \in E(C)} c(e) - \max_{e \in E(C)} c(e) \right] \geq h, \quad (2.1)$$

where the minimum is taken over all (s, t) -cuts $C(V_1, V_2)$ that separate s and t in G , and $E(C)$ is the set of edges that belong to C . In [28], it was shown that this condition is also sufficient for providing instantaneous recovery from edge failures. Moreover, it was shown that instantaneous recovery can be always achieved by using *linear* network codes. Therefore, we refer to a graph $G(V, E)$ that satisfies condition (2.1) as a *feasible* network.

C. Minimal and Simple Networks

A network $G(V, E)$ is said to be *minimal* with respect to the capacity function $c(e)$ if it satisfies the following two conditions:

1. $G(V, E)$ is a feasible network;
2. Removing an edge from G , or reducing its capacity, results in a violation of the network feasibility property.

1. Reduced Capacity Function

Networks can be made minimal by iteratively removing redundant edges and decreasing the capacity of the remaining edges. However, this approach may incur a

significant computational overhead. Accordingly, we introduce the *reduced capacity* function \bar{c} that allows to identify minimal networks in a very efficient way through the application of network flow techniques. This function is also instrumental for establishing the unique combinatorial structure of simple networks.

The reduced capacity function \bar{c} is defined as follows.

$$\bar{c}(e) = \begin{cases} 1.5 & \text{if } c(e) \geq 2; \\ 1 & \text{otherwise.} \end{cases} \quad (2.2)$$

We refer to $\bar{c}(e)$ as the *reduced capacity* of e , as opposed to the original capacity $c(e)$ of e . The following theorem establishes a connection between the feasibility of a network $G(V, E)$ with respect to the capacity function c , and the existence of a flow of value three in the network $G(V, E)$ with reduced edge capacities.

Theorem C.1. *Let $G(V, E)$ be a network, $s \in V$ be the source node, $t \in V$ be a destination node. Then, the network $G(V, E)$ is feasible with respect to the capacity function c , if and only if it admits a flow of value three with respect to the reduced capacity function \bar{c} .*

Proof. First, we show that if $G(V, E)$ is feasible, then it admits a flow of value three with respect to the reduced capacity function \bar{c} . Let C be an (s, t) -cut in $G(V, E)$, we show that the reduced capacity of this cut is at least three. The lemma will then follow from the max-flow min-cut theorem. We observe that by Equation (2.1), C contains at least two edges. If C contains exactly two edges, then both edges must be of capacity two; hence their reduced capacity is equal to 1.5, or three in total. If C contains more than three edges, then their total reduced capacity is also at least three.

Second, suppose that network $G(V, E)$ admits a flow of value three with respect

to the reduced capacities. This implies that the reduced capacity of any (s, t) -cut C in $G(V, E)$ is at least three. Since the reduced capacity of any edge is at most 1.5, the cut C has at least two edges. If C contains exactly two edges, then the reduced capacity of each edge is equal to 1.5, which implies that their original capacity is equal to 2. If C contains three edges, then the original capacity of each edge is at least one. In both cases, the conditions of Equation (2.1) is satisfied for $h = 2$. \square

The function \bar{c} can be used to verify whether a given network $G(V, E)$ is feasible with respect to a given capacity function c . This function will also serve as a building block for the algorithm that finds minimal sub-networks.

Suppose that $G(V, E)$ admits a flow of value three with respect to the reduced capacity function. Then, by the integrality property [69, Theorem 9.10], there always exists a minimum-cost flow of value three, such that $\theta(e) \in \{0, 0.5, 1, 1.5\}$ for each $e \in E$. We refer to such flow as a *half-integral* flow.

The following lemma establishes a relation between the original capacities in a minimal network and the corresponding edge flow in the network with the reduced capacities.

Lemma C.2. *Let $G(V, E)$ be a minimal network and let θ be a half-integral flow of value three in the network $G(V, E)$ with the reduced capacity function \bar{c} . Then, the following conditions hold:*

1. *For each edge $e \in E$ for which $\bar{c}(e) = 1.5$ it holds that $\theta(e) = 1.5$;*
2. *For each edge $e \in E$ for which $\bar{c}(e) = 1$ it holds that either $\theta(e) = 0.5$ or $\theta(e) = 1$.*

Proof. Suppose that there exists an edge e such that $\bar{c}(e) = 1.5$ and $\theta(e) \leq 1$. Let c' be the capacity function formed from c by reducing the capacity of e by one. Then, the

network $G(V, E)$ will be feasible with respect to c' , which contradicts the minimality assumption. Using a similar argument it can be shown that the existence of an edge e such that $\bar{c}(e) = 1$ and $\theta(e) = 0$ also contradicts the minimality of the network. \square

The next lemma shows that a minimal network $G(V, E)$ is acyclic.

Proposition C.3. *Let $G(V, E)$ be a minimal network with respect to the capacity function c . Then $G(V, E)$ does not contain cycles.*

Proof. Suppose, by way of contradiction, that $G(V, E)$ contains a cycle W . Let θ be a minimum cost flow in $G(V, E)$ with the reduced capacities. The Negative Cycle Optimality condition [69, Theorem 9.1] implies that there does not exist a cycle in $G(V, E)$ with strictly positive flow on each edge. Thus, there exists an edge $e \in W$ for which it holds that $\theta(e) = 0$, in contradiction to Lemma C.2. \square

2. Simple Networks

As mentioned in the Introduction, our goal is to establish the combinatorial structure of minimal networks. For clarity of presentation, we focus on a special class of such networks, referred to as *simple networks*. For any minimal network a corresponding simple network can be constructed through an efficient procedure.

Definition C.1 (Simple Unicast Networks). A unicast network $G(V, E)$ is said to be *simple* with respect to the capacity function c if it satisfies the following four conditions:

1. $G(V, E)$ is a minimal network with respect to c ;
2. The source node s has exactly three outgoing edges of capacity one; the destination node t has exactly three incoming edges of capacity one;
3. The degree of each node $v \notin \{s, t\}$ is exactly three;

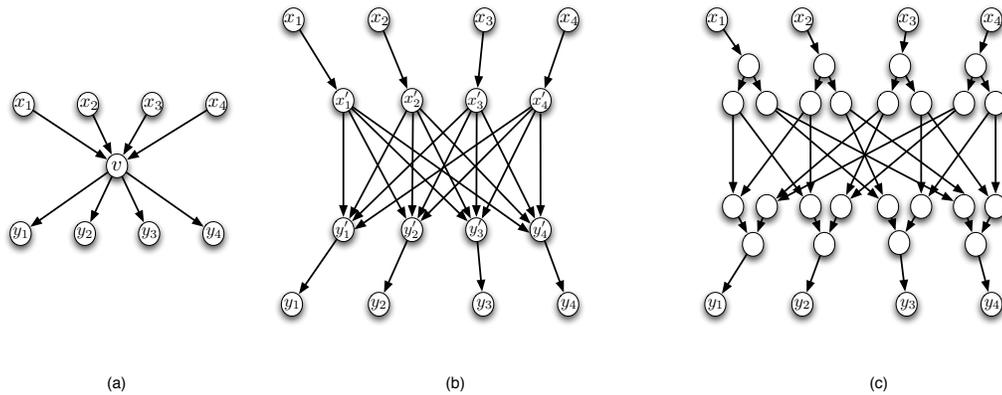


Fig. 14. (a) Node v of degree eight; (b) The intermediate step in constructing the gadget Γ_v ; (c) The final step in constructing the gadget Γ_v .

4. For every two nodes u and v , there is at most one edge in E from u to v , i.e., E does not contain parallel edges.

We proceed to describe an efficient algorithm for finding for any arbitrary network $G(V, E)$ with capacity function c an “equivalent” simple network. The transformation includes a sequence of steps that remove redundant edges and reduce the excessive edge capacities. The transformation preserves the feasibility of the graph. Moreover, any feasible network code for the simple network can be used for the original network as well, with some straightforward modifications. Our algorithm includes the following steps:

1. Add a new source node \hat{s} to G and connect it to s by three edges of capacity one. Similarly, add a new destination node \hat{t} and connect t to \hat{t} by three edges of capacity one.
2. Find a minimum cost half-integral flow θ of value three with respect to the reduced capacity function \bar{c} as defined by Equation (2.2) and assuming unit

edge costs .

3. Remove redundant edges and decrease capacities:
 - (a) Remove from $G(V, E)$ all edges e for which it holds that $\theta(e) = 0$;
 - (b) For each edge $e \in E$ for which it holds that $\theta(e) \in \{0.5, 1\}$, set $c(e) = 1$;
 - (c) For each edge $e \in E$ for which it holds that $\theta(e) = 1.5$, set $c(e) = 2$.

4. Substitute each internal node $v \in V$, $v \notin \{\hat{s}, \hat{t}\}$, in the resulting network of degree larger than three by a gadget Γ_v , constructed as follows:
 - (a) Let E_v^{in} and E_v^{out} be the incoming and outgoing edges of v , respectively. For each edge $(x, v) \in E_v^{in}$ we add a node x' to Γ_v and substitute edge (x, v) by edge (x, x') (of equal capacity). Similarly, for every edge $(v, y) \in E_v^{out}$ we add a node y' to Γ_v and substitute edge (v, y) by edge (y', y) (of equal capacity). Next, for each edge $(x, v) \in E_v^{in}$ and each edge $(v, y) \in E_v^{out}$ we connect the nodes x' and y' by edges of capacity two. Figure 14(b) depicts an example of the resulting gadget;
 - (b) Each node whose in-degree is equal to one and out-degree is more than two is substituted by a binary tree as depicted in Figure 14(c). A similar operation is performed for nodes whose out-degree is equal to one and whose in-degree is greater than one. Note that in the resulting network, the total degree of each node is at most three.

5. For each edge $e \in E$ we check whether the removing it from E results in a violation of the feasibility condition of Equation (2.1). To this end, we check whether there exists an (s, t) -flow of size three in the network $G(V, E)$ with respect to the reduced capacities. A similar procedure is performed to reduce

the capacity of each edge to the minimum possible amount while keeping the network feasible.

6. If $v \in G(V, E)$ is of degree two, then v has one incoming edge (u, v) , and one outgoing edge (v, w) . For each such node v , we substitute edges (u, v) and (v, w) by a single edge (u, w) of the same capacity and remove node v .
7. For every two nodes v and $u \in V \setminus \{s, t\}$ connected by two parallel edges $e'(v, u)$ of capacity c' , and $e''(v, u)$ of capacity c'' , we substitute the edges $e'(v, u)$ and $e''(v, u)$ by a single edge $e(v, u)$ of capacity $c' + c''$.

The purpose of Step 1 is to ensure that the source node s has exactly three outgoing edges and the destination node t has exactly three incoming edges of capacity one. In Step 2, we find a flow of minimum cost and of value three which is used in Step 3 to remove the redundant edges and decrease the capacity of other edges. The goal of Step 4 is to ensure that the degree of each node is bounded by three. The goal of Step 5 is to ensure the minimality of the resulting graph. In Step 6, we remove nodes of degree two. In Step 7, we substitute parallel edges by a single edge of larger capacity. Note that the transformation to a simple network is not unique in general.

The previous construction of a simple network can be accomplished in $O(V^2)$ time. Indeed, Step 1 requires linear time $O(V)$, while steps 2 and 3 require $O(E)$ time. We also note that, after Step 2, the network contains only $O(V)$ edges. This implies that the computational complexity of Steps 5, 6, and 7 is $O(V)$. Finally, Step 4 requires $O(V^2)$ time.

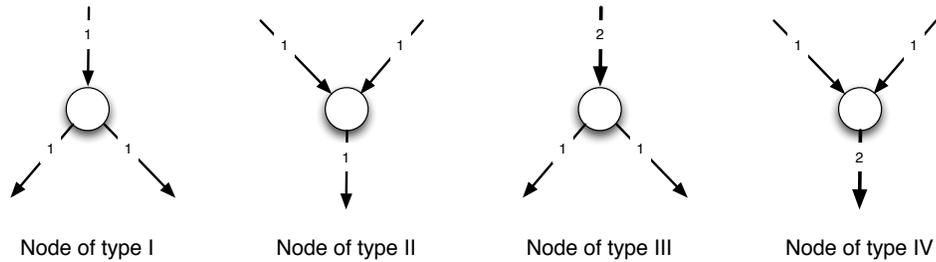


Fig. 15. The four possible types of nodes in a simple unicast network.

D. Structure of Simple Networks

1. Node Properties of Simple Unicast Networks

Let $G(V, E)$ with source node s and destination node t be a simple network with respect to the capacity function c . We say that a node $v \in V$ is of Type I if it has one incoming edge and two outgoing edges, all of capacity one; of Type II if it has two incoming edges and one outgoing edge, all of capacity one; of Type III if it has one incoming edge of capacity two and two outgoing edges of capacity one; of Type IV if it has two incoming edges of capacity one and one outgoing edge of capacity two. Figure 15 depicts nodes of Types I, II, III, and IV.

The next theorem proves that each node $v \in V \setminus \{s, t\}$ in a simple network is either of Type I, II, III, or IV.

Lemma D.1. *Let $G(V, E)$ be a simple network. Then, each node $v \in V \setminus \{s, t\}$ is of the type I, II, III, or IV.*

Proof. Since $G(V, E)$ is a feasible graph, Theorem C.1 implies that there exists a flow of value three in $G(V, E)$ with reduced edge capacities. Let θ be a minimum cost half-integral flow in $G(V, E)$ with respect to \bar{c} . Let $v \in V \setminus \{s, t\}$ be an internal

node of the network. Since the network is simple, the total degree of v is equal to three. Assume first that v has one incoming edge e and two outgoing edges. If the capacity of e is equal to one then, by Lemma C.2, it holds that $\theta(e) \in \{0.5, 1\}$. It is easy to verify that $\theta(e) = 1$, otherwise one of the outgoing edges has zero flow, in contradiction to the minimality of $G(V, E)$. In this case, the flow on the outgoing edges of v is equal to 0.5 and by Lemma C.2 their capacity is equal to one, which implies that v is a Type I node. If the capacity of e is equal to two, then, by Lemma C.2, it must be the case that $\theta(e) = 1.5$. Then, the outgoing edges of v have flow of value 0.5 and 1. Thus, by the same lemma, the capacity of these edges is equal to one, which implies that v is a Type III node.

By using a similar argument, we can show that if v has two incoming edges and one out-going edge than it is either of Type II or IV. \square

2. Residual Graphs and Residual Cycles

Let $G(V, E)$ be a simple network, and let θ be a minimum cost half-integral flow of value three in $G(V, E)$ with respect to the reduced capacity function \bar{c} . We define the set $\hat{E} \subseteq E$ as follows:

$$\hat{E} = \{e \in E \mid \bar{c}(e) = 1 \text{ and } \theta(e) = 0.5\}. \quad (2.3)$$

Note that \hat{E} includes every edge $e \in E$ for which $\theta(e) \leq \bar{c}(e)$, i.e., edges that have residual capacity and can take up more flow. Let E_1 be a subset of \hat{E} . We define the subset E_2 to be:

$$E_2 = \{e \in E \mid \theta(e) = 1.5\} \cup \{e \in \hat{E} \mid e \notin E_1\} \quad (2.4)$$

Note that the set E_2 depends on the choice of the set E_1 . Intuitively, the set E_2 includes edges for which the amount of flow can be reduced by adding more flow to

edges in E_1 .

Definition D.1 (Residual Graph). Let E_1 be a subset of \hat{E} . Then, the *residual graph* $G_{E_1}(\theta)$ of $G(V, E)$ is formed from $G(V, E)$ by reversing all edges in $E \setminus E_1$.

Let W be a cycle in the residual graph. Since the graph network is acyclic, W must contain at least one edge in E_1 . By augmenting the flow θ along W we can increase the flow on edges in $W \cap E_1$ and decrease the flow on other edges of W . We refer to a cycle in the residual graph that includes an edge in E_2 as a *residual cycle*. The existence of a residual cycle implies that the amount of the flow on some edge in E_2 can be reduced. The following lemma shows that if $G(V, E)$ is minimal, then $G_{E_1}(\theta)$ does not contain a residual cycle.

Lemma D.2. *Let $G(V, E)$ be a simple network, let θ be a minimum cost half-integral flow of value three in $G(V, E)$ with respect to the capacity function \bar{c} , and let E_1 be a subset of \hat{E} . Then, the residual graph $G_{E_1}(\theta)$ does not contain a residual cycle.*

Proof. Suppose, by contradiction, that there exists a residual cycle W in $G_{E_1}(\theta)$. Such cycle must include at least one edge $e \in E_2$. Let θ' be the flow obtained by augmenting θ along W , i.e.,

$$\theta'(e) = \begin{cases} \theta(e) + 0.5 & \text{if } e \in E_1 \cap W; \\ \theta(e) - 0.5 & \text{if } e \in W \setminus E_1; \\ \theta(e) & \text{otherwise.} \end{cases}$$

It is easy to verify that θ' is a feasible half-integer flow of value three in $G(V, E)$ with respect to \bar{c} . Let e be an edge of the residual cycle that belongs to E_2 . Then one of the two following conditions hold:

1. $\bar{c}(e) = 1$ and $\theta'_e = 0$;
2. $\bar{c}(e) = 1.5$ and $\theta'_e = 1$.

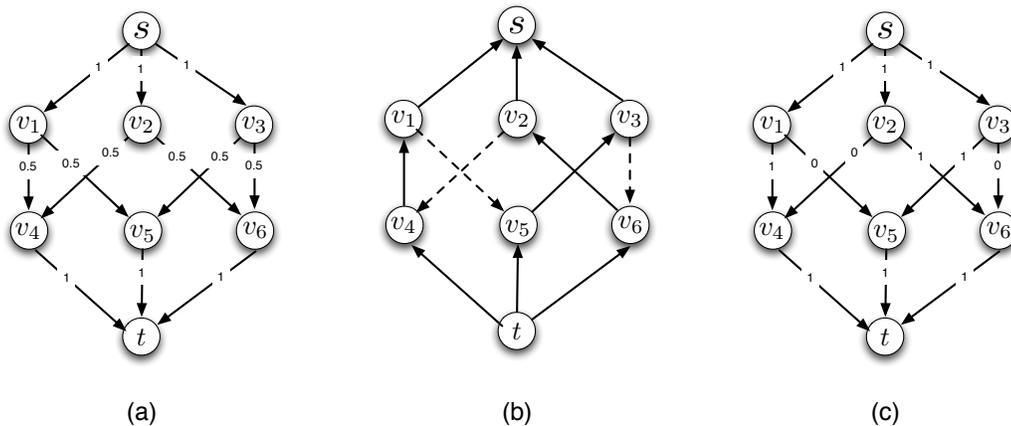


Fig. 16. (a) A graph $G(V, E)$ with edges of unit capacity and a flow θ of value three. Each edge $e \in E$ is labeled with the amount of flow $\theta(e)$ it carries. $\hat{E} = \{(v_1, v_4), (v_1, v_5), (v_2, v_4), (v_2, v_6), (v_3, v_5), (v_3, v_6)\}$. (b) Residual graph for $E_1 = \{(v_1, v_5), (v_2, v_4), (v_3, v_6)\}$. The graph contains a residual cycle $W = \{v_1, v_5, v_3, v_6, v_2, v_4, v_1\}$. (c) The flow θ' obtained from θ by augmenting along cycle W . Note that edges (v_1, v_5) , (v_2, v_4) , and (v_3, v_6) are redundant and can be removed from the network without violating its feasibility.

By Lemma C.2, this contradicts the minimality of $G(V, E)$. \square

Figure 16 provides an example of a non-minimal network, the construction of a residual graph, and the result of augmenting flow θ along a residual cycle.

3. Block Decomposition

Let $G(V, E)$ be a simple unicast network with at least one node other than s and t . We show that any such network can be decomposed into a set of blocks of Types A , B , and C , as depicted in Figure 17(a).

Theorem D.3. *Let $G(V, E)$ be a simple unicast network with $|V| > 2$. Then, $G(V, E)$*

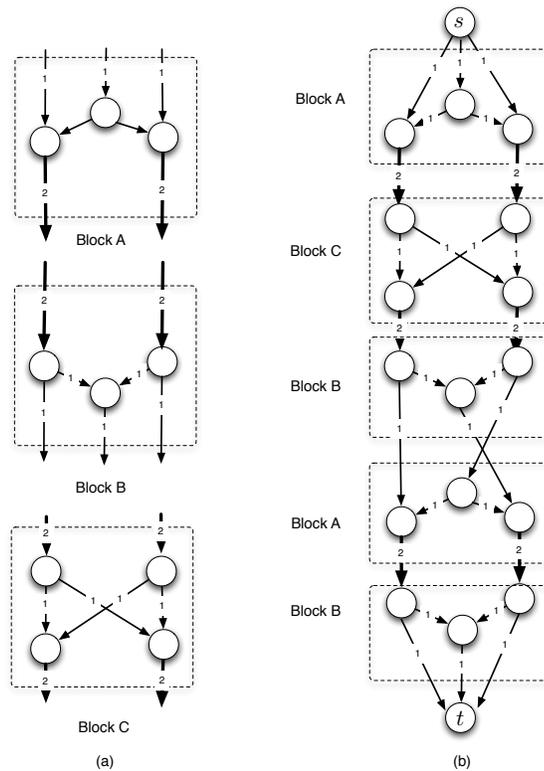


Fig. 17. (a) The three basic building blocks of types A, B and C, for simple unicast networks. (b) An example of the block decomposition of a simple unicast network.

can be decomposed into a sequence of the blocks of type A, B or C which are depicted in Figure 17(a). The blocks can be appear in an arbitrary order, subject to the following rules:

1. The first block is of Type A, and s is incident to its input edges;
2. The last block is of Type B, and the destination node t is incident its output edges;
3. A block of Type A is always followed by a block of Type B or Type C;
4. A block of Type B is either followed by a block of Type A or connected to the

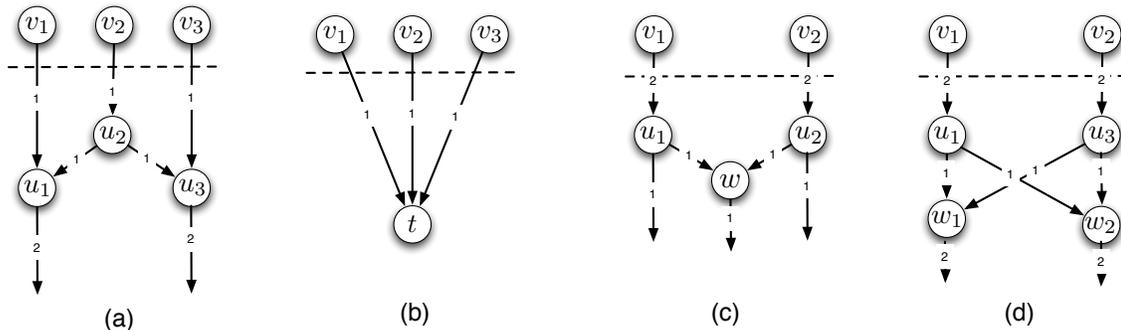


Fig. 18. (a) and (b) Examples of cuts of Type 1; (c) and (d) Examples of cuts of Type 2.

destination;

5. A block of Type C is followed by a block of Type B or Type C.

Let $G(V, E)$ be a simple network and $C(V_1, V_2)$ an (s, t) -cut of $G(V, E)$. We denote by $E(C) \subseteq E$ the set of edges that belong to C . We say that $C(V_1, V_2)$ is a cut of *Type 1* if $E(C)$ includes three edges of unit capacity. A cut $C(V_1, V_2)$ is said to be of *Type 2* if it includes two edges of capacity two. Figure 18 shows examples of cuts of Types 1 and 2.

In what follows we prove two lemmas that capture the properties of simple networks. The first lemma implies that any cut of Type 1 is followed by either the destination node t or a block of Type A.

Lemma D.4. *Let $G(V, E)$ be a simple network. Let $C = (V_1, V_2)$ be a cut of Type 1 in $G(V, E)$, i.e., $E(C)$ contains three unit capacity edges $e_1(v_1, u_1)$, $e_2(v_2, u_2)$, and $e_3(v_3, u_3)$, originating at V_1 and ending in V_2 . Then, either:*

- $u_1 = u_2 = u_3 = t$,

- or, one of the nodes u_1 , u_2 , or u_3 is of Type I, while two other nodes are of Type IV. Moreover, the node of Type I is adjacent to the two other nodes as depicted in Figure 18(a).

The next lemma implies that any cut of Type 2 is followed by either a block of Type B or Type C.

Lemma D.5. *Let $G(V, E)$ be a simple network. Let $C = (V_1, V_2)$ be a cut of Type 2 in $G(V, E)$, i.e., $E(C)$ includes two edges $e_1(v_1, u_1)$, $e_2(v_2, u_2)$, each one of them is of capacity two. Then, there exist*

- either a Type II node $w \in V_2$, and two edges (u_1, w) and (u_2, w) of unit capacities,
- or two nodes w_1 and w_2 of Type IV and four edges (u_1, w_1) , (u_1, w_2) , (u_2, w_1) , and (u_2, w_2) of unit capacity 1, as depicted in Figures 18(c) and (d).

The proof of Lemma D.4 appears in Section 4, while the proof of Lemma D.5 appears in Appendix A. It is easy to verify that Lemmas D.4 and D.5 are sufficient for proving the correctness of Theorem D.3.

4. Proof of Lemma D.4

Let $G(V, E)$ be a simple network, and let θ be a flow of value three with respect to the reduced edge capacities \bar{c} . Also, let $C(V_1, V_2)$ be an (s, t) -cut $C(V_1, V_2)$ of Type 1 in $G(V, E)$. We denote by $E(C) = \{(v_1, u_1), (v_2, u_2), (v_3, u_3)\}$ the set of edges in to C .

First, we observe that each of the nodes u_i , $i = 1, 2, 3$, is either a destination node or a node of Type I or IV. Indeed, u_i cannot be of Type III because the capacity of edge (v_i, u_i) is equal to one. Also, for the flow θ and for each edge $e_i = (v_i, u_i) \in E(C)$, it must hold that $\theta(e_i) = 1$, which implies that the nodes u_i cannot be of Type II.

Next, we assume that $t \notin \{u_1, u_2, u_3\}$ and show, by contradiction, that at most one of the nodes u_1 , u_2 , and u_3 is of Type I. We consider two cases. In the first case, the three nodes u_1 , u_2 , and u_3 are Type I nodes. In the second case, two of the nodes are of Type I, while the other node is of Type IV.

Case 1. Suppose that all three nodes u_1 , u_2 , and u_3 are of Type I. In this case, all the outgoing edges of u_1 , u_2 , and u_3 belong to \hat{E} (defined by Equation (2.3)). Since the in-degree of any node of $G \setminus \{s, t\}$ is either 1 or 2, we can always pick three edges e^1 , e^2 , and e^3 such that, for $i = 1, 2, 3$, e^i is an outgoing edge of u_i , and e^1 , e^2 , and e^3 are independent, i.e., there does not exist a node v which is incident to any two edges e^i, e^j ($i \neq j$). For example, in Figure 16, we can chose $e^1 = (v_1, v_5)$, $e^2 = (v_2, v_4)$, and $e^3 = (v_3, v_6)$.

Let $E_1 = \{e^1, e^2, e^3\}$ and let E_2 be the set defined by Equation (2.4). Let $G_{E_1}(\theta)$ be the residual graph of $G(V, E)$ with respect to E_1 , and let G' be the subgraph of $G_{E_1}(\theta)$ induced by the nodes in V_2 . We observe that each node in G' has an out-degree at least one due to the following facts:

1. None of the edges incident to the destination node t belongs to E_1 . Hence, t has three outgoing edges in G' ;
2. Any node in $G' \setminus \{t\}$ which is not a head or a tail of an edge in E_1 , has at least one incoming edge that does not belong to E_1 . Hence, its out-degree in G' is at least one.
3. Nodes u_1, u_2 and u_3 have outgoing edges e^1, e^2 and e^3 , respectively. Since these edges belong to E_1 , they have the same direction as in the original graph G .
4. Lemma C.2 implies that any head node v of an edge in E_1 is either of Type II or IV. In both cases, v has an outgoing edge that does not belong to E_1 . Hence,

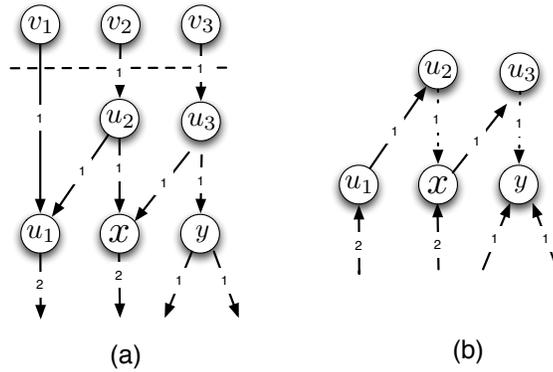


Fig. 19. (a) An example of a Type 1 cut with two nodes of Type I and one node of Type IV. (b) The corresponding graph G' with $E_1 = \{(u_2, x), (u_3, y)\}$.

its out-degree in G' is at least one.

We conclude that G' contains a cycle. Such a cycle must include at least one edge e^i , for some $i \in \{1, 2, 3\}$, because $G' \setminus \{e^1, e^2, e^3\}$ is an acyclic graph. This implies that this cycle includes at least one edge in E_2 , which is the edge incident to u_i . Therefore, G' and, in turn, $G_{E_1}(\theta)$, include a residual cycle, which by Lemma D.2 contradicts the minimality of $G(V, E)$.

Figure 16 depicts a reduced capacity network with a cut $C(s, E \setminus s)$ of Type I in which all the nodes u_1 , u_2 and u_3 are of Type I. The corresponding residual graph contains a residual cycle, and thus the original unicast network is not minimal.

Case 2. In this case, one of the nodes, say u_1 is of Type IV, while the two other nodes u_2 and u_3 are of Type I. We choose $E_1 = \{e^2, e^3\} \subseteq \hat{E}$ such that e^2 is an outgoing edge of u_2 , e^3 is an outgoing edge of u_3 , neither one of them is incident to u_1 , and no node is incident to both e^2 and e^3 . It can be verified that such choice is always possible. Let E_2 be the set defined by Equation (2.4). An example of this case is depicted in Figure 19.

Following the same argument as above, we denote by $G_{E_1}(\theta)$ the residual graph of $G(V, E)$ with respect to E_1 , and by G' the subgraph of $G_{E_1}(\theta)$ induced by nodes in V_2 . It is easy to verify that each node in G' has out-degree at least 1, hence G' includes a cycle. Such a cycle must include either e^2 or e^3 , or both. This implies that the cycle includes an edge in E_2 , which, by Lemma D.2, contradicts the minimality of the network.

Next, we prove that if one of the nodes in $\{u_1, u_2, u_3\}$ is of Type I, then it must be adjacent to the two other nodes which are of Type IV. We assume, without loss of generality, that u_1 is a Type I node and u_2 and u_3 are of Type IV. Suppose, by way contradiction, that u_1 is not adjacent to at least one of the nodes u_2 and u_3 . We denote by e an outgoing edge of u_1 which is not adjacent to u_2 and u_3 . Let $E_1 = \{e\}$ and let E_2 be set defined by Equation (2.4).

Let $G_{E_1}(\theta)$ be the residual graph of $G(V, E)$ with respect to E_1 , and G' the subgraph of $G_{E_1}(\theta)$ induced by the nodes in V_2 . It is easy to verify that the out-degree of each node in G' is at least one, hence G' includes a cycle. Such a cycle must include the edge e , and, in turn, one edge in E_2 which is incident to u_1 . As a result, G' and, in turn, $G_{E_1}(\theta)$ include a residual cycle, in contradiction to Lemma D.2.

It can be proven, by using a similar argument that if one of the nodes $\{u_1, u_2, u_3\}$ is identical to t , then all other nodes in $\{u_1, u_2, u_3\}$ are identical to t , otherwise at least one of the nodes $\{u_1, u_2, u_3\}$ must be of Type I, in contradiction to the minimality of the original network.

We have shown that all cases other than those mentioned in the condition of the Lemma contradict the minimality of the coding network, hence the result follows.

E. Network Codes for Simple Networks

In this section, we present a robust network code over $GF(2)$ for the network consisting of a simple network G , source node s and destination node t , and prove its correctness.

As shown in the previous section, a simple unicast network consists of a sequence of blocks of types A , B , and C , depicted in Figure 17(a). The source node s has three outgoing edges of capacity one, connected to a block of Type A . We denote by $p_1, p_2 \in GF(2)$ the two packets that the source node has to transmit to the destination t during the current round. Our network code can be specified as follows. First, the source node sends packets p_1 , p_2 , and $p_1 + p_2$ on its outgoing arcs. Second, the encoding functions for the arcs of blocks A , B , and C are depicted in Figure 20. The figure shows for each edge e_i of capacity one the corresponding arc a_i , and for each edge e_i of capacity two the corresponding arcs a_i^1 and a_i^2 . We choose the notation such that if edge e_i of block X coincides with edge e_j of block Y , then arcs a_i^1 and a_i^2 of block X coincide with arcs a_j^1 and a_j^2 of block Y , respectively. Note that all the arcs that belong to the blocks of Types A and C just forward their incoming packets, while each block of Type B has two encoding nodes.

We proceed to prove that the network code described above is robust, i.e., the destination node can always recover the original packets even if one of the edges in the original network fails. Consider a simple network $G(V, E)$ that contains n blocks of Type A . Recall that each block of Type A is either followed by a block of Type B , or by several blocks of Type C , which, in turn, are followed by a block of Type B . We denote by A_i , $1 \leq i \leq n$, the i^{th} block of Type A from the source. We also denote by B_i , $1 \leq i \leq n$, the i^{th} block of Type B from the source such that B_1 is the first block of Type B after A_1 .

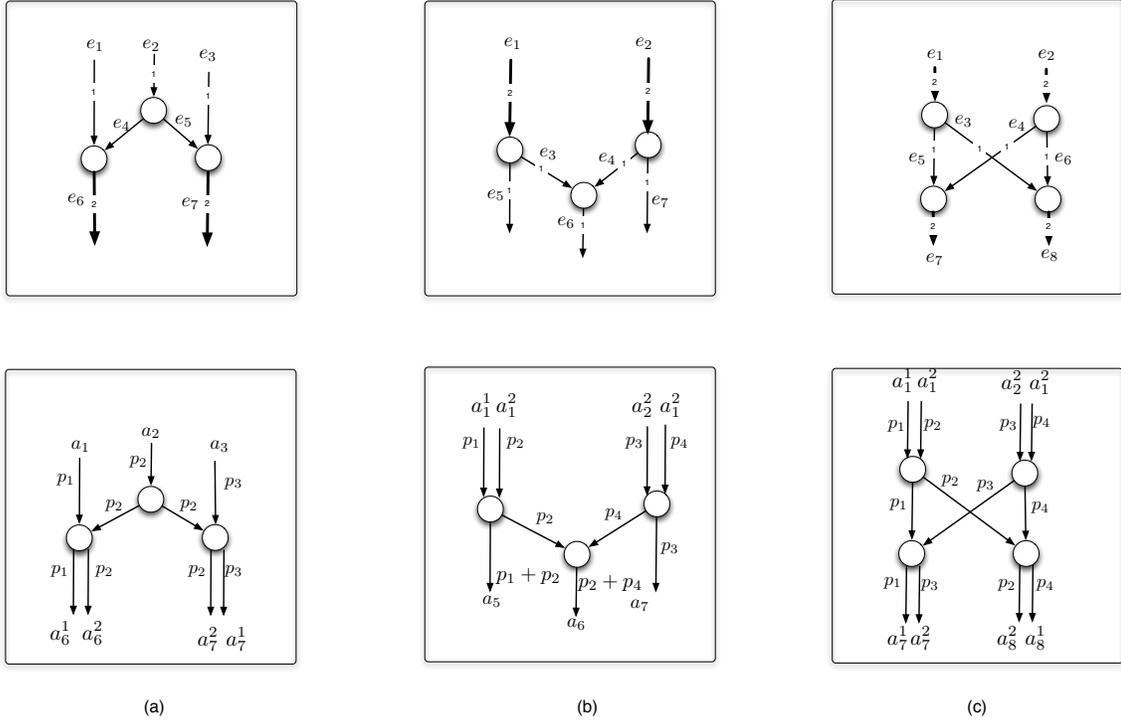


Fig. 20. Robust network code for simple unicast networks: (a) Encoding for blocks of Type A; (b) Encoding for blocks of Type B; (c) Encoding for blocks of Type C.

We define $I_{A_i} := (p_{a_1}^i, p_{a_2}^i, p_{a_3}^i)$ to be the vector of packets entering block A_i , where $p_{a_1}^i, p_{a_2}^i$ and $p_{a_3}^i$ are the packets carried by arcs a_1, a_2 and a_3 of block A_i , respectively. We also define $O_{A_i} = (p_{a_6}^i, p_{a_6}^i, p_{a_7}^i, p_{a_7}^i)$ to be the vector of packets leaving block A_i , where $p_{a_6}^i, p_{a_6}^i, p_{a_7}^i$, and $p_{a_7}^i$ are the packets carried by arcs a_6^1, a_6^2, a_7^1 and a_7^2 of block A_i , respectively. Similarly, we let $I_{B_i} = (p_{a_1}^i, p_{a_1}^i, p_{a_2}^i, p_{a_2}^i)$ and $O_{B_i} = (p_{a_5}^i, p_{a_6}^i, p_{a_7}^i)$ be the vectors of packets leaving block B_i , respectively. Recall that if edge e fails, then the encoding function of each arc $a \in A(e)$ that corresponds to e is identically zero, i.e., $f_a \equiv 0$.

Lemma E.1. Consider a block $A_i, i \in \{1, \dots, n\}$. Suppose that there are no failures

in A_i , B_i , or the blocks of Type C located between A_i and B_i . Suppose also that the input vector I_{A_i} of A_i is a permutation of $(p_1, p_2, p_1 + p_2)$. Then, the output vector O_{B_i} of B_i is a permutation of $(p_1, p_2, p_1 + p_2)$.

Proof. The proof immediately follows from the network code for blocks of types A, B, and C, depicted in Figure 20. \square

From Lemma E.1 it follows that if there are no failures in the blocks located between s and B_i , then the output vector O_{B_i} of B_i is a permutation of $(p_1, p_2, p_1 + p_2)$. The following lemma characterizes the output of block B_i in the case of an edge failure in the blocks A_i , B_i , or a block of Type C located between A_i and B_i .

Lemma E.2. *Consider a block A_i , $i \in \{1, \dots, n\}$. Suppose that there is an edge failure in A_i , B_i , or the blocks of Type C located between A_i and B_i . Suppose also that the input vector I_{A_i} of A_i is a permutation of $(p_1, p_2, p_1 + p_2)$. Then, one of the following holds:*

1. *The output vector O_{B_i} of B_i is a permutation of $(p_1, p_2, p_1 + p_2)$;*
2. *The output vector O_{B_i} of B_i includes two distinct elements from the set $\{p_1, p_2, p_1 + p_2\}$ and a zero;*
3. *The output vector O_{B_i} of B_i includes two distinct elements from the set $\{p_1, p_2, p_1 + p_2\}$, one of which appearing twice.*

Proof. First, we consider the case in which the failed edge belongs to block A_i . Let $O_{A_i} = (p_{a_6}^i, p_{a_6}^i, p_{a_7}^i, p_{a_7}^i)$ be the output vector of block A_i , and $I_{B_i} = (p_{a_1}^i, p_{a_1}^i, p_{a_2}^i, p_{a_2}^i)$ the input vector of block B_i . For all failure scenarios in block A_i , one of the following conditions holds:

1. *Vector $(p_{a_6}^i, p_{a_6}^i, p_{a_7}^i, p_{a_7}^i)$ includes two distinct elements from the set $\{p_1, p_2, p_1 + p_2\}$ and two zeros;*

2. Vector $(p_{a_6^i}^i, p_{a_6^2}^i, p_{a_7^2}^i, p_{a_7^1}^i)$ includes three distinct elements from the set $\{p_1, p_2, p_1 + p_2\}$ and one zero;
3. Packets $p_{a_6^1}^i$ and $p_{a_6^2}^i$ correspond to two distinct elements from the set $\{p_1, p_2, p_1 + p_2\}$, $p_{a_7^2}^i = p_{a_6^2}^i$, and $p_{a_7^1}^i$ is zero;
4. Packets $p_{a_7^1}^i$ and $p_{a_7^2}^i$ correspond to two distinct elements from the set $\{p_1, p_2, p_1 + p_2\}$, $p_{a_6^2}^i = p_{a_7^2}^i$, and $p_{a_6^1}^i$ is zero.

Due to the structure and the form of the encoding functions of the blocks of Type C , it can be seen that the same condition holds if we substitute packets $p_{a_6^1}^i, p_{a_6^2}^i, p_{a_7^2}^i, p_{a_7^1}^i$ by input packets $(p_{a_1^i}^i, p_{a_1^2}^i, p_{a_2^2}^i, p_{a_2^1}^i)$ of block B_i , respectively. Then, the output vector O_{B_i} of B_i satisfies the condition of the lemma.

Next, we consider the case in which the failed edge belongs to one of the blocks of Type C located between A_i and B_i . Let \hat{C} be a block with a faulty edge. We denote by $p_{a_1^1}, p_{a_1^2}, p_{a_2^2},$ and $p_{a_2^1}$ the packets that are incoming to arcs $a_1^1, a_1^2, a_2^2,$ and a_2^1 of \hat{C} , respectively. Since the preceding blocks of \hat{C} do not contain failed edges it holds that the vector $(p_{a_1^1}, p_{a_1^2}, p_{a_2^2}, p_{a_2^1})$ contains three distinct packets from $(p_1, p_2, p_1 + p_2)$ and $p_{a_1^2} = p_{a_2^2}$. Let $p_{a_7^1}, p_{a_7^2}, p_{a_8^2},$ and $p_{a_8^1}$ the packets carried by the output arcs $a_7^1, a_7^2, a_8^2,$ and a_8^1 of \hat{C} . It is easy to verify that these packets satisfy one of the following conditions:

1. Vector $(p_{a_7^1}, p_{a_7^2}, p_{a_8^2}, p_{a_8^1})$ includes two distinct elements from the set $\{p_1, p_2, p_1 + p_2\}$ and two zeros;
2. Vector $(p_{a_7^1}, p_{a_7^2}, p_{a_8^2}, p_{a_8^1})$ includes three distinct elements from the set $\{p_1, p_2, p_1 + p_2\}$ and one zero.

Due to the structure and the form of the encoding functions of blocks of Type C , the same condition holds if we substitute packets $p_{a_7^1}, p_{a_7^2}, p_{a_8^2}, p_{a_8^1}$ by input packets

$(p_{a_1}^i, p_{a_2}^i, p_{a_2}^i, p_{a_1}^i)$ of block B_i , respectively. It can be verified that the output vector O_{B_i} of B_i satisfies the condition of the lemma.

Finally, we consider the case in which the failed edge belongs to block B_i , and let $(p_{a_1}^i, p_{a_1}^i, p_{a_2}^i, p_{a_2}^i)$ be the input packets of block B_i . Since the preceding blocks of \hat{C} do not contain failed edges, it holds that the vector $(p_{a_1}^i, p_{a_1}^i, p_{a_2}^i, p_{a_2}^i)$ contains three distinct packets from the set $\{p_1, p_2, p_1 + p_2\}$ and $p_{a_1}^i = p_{a_2}^i$. It can be verified that the output vector O_{B_i} of B_i satisfies the condition of the lemma. \square

Lemma E.3. *Consider a block A_i , $i \in \{1, \dots, n\}$. Suppose that there is no edge failure in A_i , B_i , or in the blocks of Type C located between A_i and B_i . Suppose also that the output vector $I_{B_{i-1}}$ of the previous block B_{i-1} satisfies the conditions of Lemma E.2. Then the output vector I_{B_i} of block B_i also satisfies the conditions of that lemma.*

Proof. First, we note that if the output vector $O_{B_{i-1}}$ of B_{i-1} is a permutation of $(p_1, p_2, p_1 + p_2)$, then by Lemma E.1 the same holds for the input vector I_{B_i} of B_i .

Next, we consider the case in which the output vector $O_{B_{i-1}}$ of B_{i-1} includes two distinct elements of $(p_1, p_2, p_1 + p_2)$ and a zero. Note that this case is equivalent to the failure of one of the incoming edges of block A_i , hence by Lemma E.2 the output vector I_{B_i} of B_i satisfies the conditions of the lemma.

Finally, we consider the case in which the output vector $O_{B_{i-1}}$ of B_{i-1} includes two distinct elements of $(p_1, p_2, p_1 + p_2)$, one of which appears twice. In this case the output vector $O_{A_i} = (p_{a_6}^i, p_{a_6}^i, p_{a_7}^i, p_{a_7}^i)$ of A_i satisfies the following conditions:

1. $(p_{a_6}^i, p_{a_6}^i, p_{a_7}^i, p_{a_7}^i)$ includes two distinct elements of the set $\{p_1, p_2, p_1 + p_2\}$;
2. $p_{a_6}^i = p_{a_7}^i$;
3. $(p_{a_6}^i, p_{a_6}^i, p_{a_7}^i, p_{a_7}^i)$ does not any contain zero packets.

The structure of the blocks of Type C implies that the same holds if the packets O_{A_i} are substituted by the packets from $I_{B_i} = (p_{a_1}^i, p_{a_1}^i, p_{a_2}^i, p_{a_2}^i)$. Thus, there are two possible cases

- Case 1: $p_{a_1}^i = p_{a_1}^i = p_{a_2}^i, p_{a_2}^i \neq p_{a_2}^i$,
- Case 2: $p_{a_2}^i = p_{a_2}^i = p_{a_1}^i, p_{a_1}^i \neq p_{a_1}^i$.

In both cases, all of the packets $p_{a_1}^i, p_{a_1}^i, p_{a_2}^i, p_{a_2}^i$ are non-zero. It can be verified that the encoding functions of block B ensure that the output vector I_{B_i} of B_i satisfies the conditions of the lemma. \square

We conclude with the following theorem:

Theorem E.4. *The proposed network code depicted in Figure 20 guarantees an instantaneous recovery from single edge failures.*

Proof. Follows from lemmas E.1, E.2, and E.3. \square

Network coding algorithm. The algorithm for finding a feasible network code includes the following steps. First, we identify the corresponding simple network using the algorithm described in Section 2. Then, we visit the nodes of the graph in topological order and group them into blocks of types A , B , and C . Next, for each block we apply the coding scheme as described above. Finally, we determine the network code for the original network. The computation complexity of the algorithm is $O(V^2)$.

F. Minimizing the Required Amount of Network Resources

In this section we present an efficient algorithm for capacity allocation for robust unicast networks. Our algorithm takes advantage of the properties of minimum networks,

established in the previous section. In the general case, the capacity reservation problem can be formulated as follows. Consider a directed graph $G(V, E)$ with a source node $s \in V$, a destination node $t \in V$ and where each edge $e \in E$ is associated with two parameters:

1. c_e : the capacity of e , i.e., the upper bound on the number of packets that can be transmitted by c_e at each communication round;
2. m_e : the cost of reserving a unit capacity on edge e .

A reservation x in the graph $G(V, E)$ is a map $x : E \rightarrow \{0, 1, \dots\}$, that assigns to each edge e the non-negative integer value x_e and satisfies the following conditions

1. The reservation x_e on each edge cannot exceed its capacity, i.e., $x_e \leq c_e$;
2. For every (s, t) -cut C that separates nodes s and t it must hold that:

$$\sum_{e \in E(C)} c(e) - \max_{e \in E(C)} c(e) \geq h. \quad (2.5)$$

The problem consists of finding an optimum reservation \hat{x} that minimizes the total cost $\sum_{e \in E} x_e \cdot m_e$.

The problem of efficient allocation of network resources for coding networks has been considered in [43]. The approach presented therein is based on linear programming techniques and does not provide provable performance guarantees for integral capacity reservations. Resilient capacity reservations has been addressed in [70, 71]; however, the case of $h = 2$ has not been addressed. In [70], it was shown that the general version of this problem is NP-hard. In what follows, we present a simple algorithm that finds the capacity reservation scheme whose cost is at most two times more than the optimum for the special case of $h = 2$. The algorithm is a variation of the algorithm presented in Section 2 and includes the following steps:

1. Find a minimum cost integral flow θ of value three with respect to the reduced capacity function \bar{c} (as defined by Equation (2.2)) and with respect to the edge costs $\{m_e \mid e \in E\}$.
2. For each edge $e \in E$ set $x(e) \leftarrow \lceil \theta(e) \rceil$.

In the following theorem we show that the resulting reservation is 2-optimal.

Theorem F.1. *The algorithm above provides a capacity reservation vector whose cost is at most two times more than the optimal one.*

Proof. Let $\hat{x}(e)$, $e \in E$ be the optimal capacity reservation scheme and let OPT be the cost of \hat{x} . We define the reduced capacity function \hat{c} as follows:

$$\hat{c}(e) = \begin{cases} 1.5 & \text{if } \hat{x}(e) = 2; \\ 1 & \text{if } \hat{x}(e) = 1; \\ 0 & \text{otherwise.} \end{cases} \quad (2.6)$$

Note that, since $\hat{c}_e \leq \hat{x}_e$, it holds that $\sum_{e \in E} \hat{c}_e \cdot m_e \leq \sum_{e \in E} \hat{x}_e \cdot m_e = OPT$. Since \hat{x} is a feasible reservation vector, Theorem C.1 implies that the reduced capacity function admits a flow $\hat{\theta}$ of value three from the source to the destination. Moreover, the cost of this flow is at most $\sum_{e \in E} \hat{c}_e \cdot m_e \leq OPT$.

We note that since $\hat{c}_e \leq \hat{x}_e$, the cost of the flow θ found by the algorithm is less than that of $\hat{\theta}$. Since θ is a half-integer flow and $x(e) = \lceil \theta(e) \rceil$, we conclude that

$$\sum_{e \in E} x_e \cdot m_e \leq 2 \cdot \sum_{e \in E} \theta_e \cdot m_e \leq 2 \cdot \sum_{e \in E} \hat{\theta}_e \cdot m_e \leq 2 \cdot OPT.$$

□

CHAPTER III

SECURE NETWORK CODING

FOR WIRETAP NETWORKS OF TYPE II

We consider in this chapter the problem of securing a multicast network against a wiretapper that can intercept the packets on a limited number of arbitrary network edges of its choice. We assume that the network employs network coding to simultaneously deliver the packets available at the source to all the receivers. We show that this problem can be looked at as a network generalization of the wiretap channel of Type II introduced in a seminal paper by Ozarow and Wyner. In particular, we show that the transmitted information can be secured by using the Ozarow-Wyner approach of coset coding at the source on top of the existing network code. This way, we easily recover many important results available in the literature on secure network coding. Moreover, we derive new bounds on the required alphabet size that are independent of the number of edges in the network, and devise an algorithm for the construction of secure network codes. In addition, we look at the dual problem and analyze the amount of information that can be gained by the wiretapper as a function of the number of wiretapped edges. The results presented in this chapter have appeared in [16, 17].

A. Introduction

Consider a communication network represented as a directed graph $G = (V, E)$ with unit capacity edges and an information source S that multicasts information to t receivers R_1, \dots, R_t located at distinct nodes. Assume that the minimum size of a cut that separates the source and each receiver node is n . It is known that a multicast rate of n is achievable by using a linear network coding scheme [?, 72]. In

this chapter, we focus on secure multicast connections in the presence of a wiretapper that can access data on a limited number of edges of its choice. Our primary goal is to design a network coding scheme that delivers data at maximum rate to all the destinations, and does not reveal any information about the transmitted message to the wiretapper.

The problem of making a linear network code information-theoretically secure in the presence of a wiretapper that can look at a bounded number μ of network edges was first studied by Cai and Yeung in [15]. They considered directed graphs, and constructed codes over an alphabet with at least $\binom{|E|}{\mu}$ elements which can support a secure multicast rate of up to $n - \mu$. In [44], they proved that these codes use the minimum amount of randomness required to achieve the security constraint. However, the algorithm due to [15] has high computational complexity and requires a very large field size that is exponential in the number of wiretapped edges. Feldman *et al.* derived trade-offs between security, code alphabet size, and multicast rate of secure linear network coding schemes in [45] by using ideas from secret sharing and abstracting the network topology.

In a network where the min-cut value between the source and each receiver node is n and where there is an eavesdropper who can access up to μ edges of his choice, we introduce a coding at source scheme that ensures information-theoretic security based on the Ozarow-Wyner wiretap channel of Type II, introduced in [18] and [19]. In this channel, the source transmits n symbols to the receiver and a wiretapper can access any μ of those symbols. Ozarow and Wyner showed that the maximum number of symbols, denoted k , that the source can securely communicate to the receiver is equal to $n - \mu$. Furthermore, they proposed a coding scheme that achieves this rate. how to encode the k source symbols into the n channel symbols for secure transmission. Clearly, if the n channel symbols are multicast over a network using a routing scheme,

the k source symbols remain secure in the presence of a wiretapper with access to any μ edges. We will illustrate later that this is not necessarily the case when network coding is used. Nevertheless, we will show that a network code based on the Ozarow-Wyner scheme that preserves security of the k source symbols, which are coded into the n multicast symbols, can be designed over a sufficiently large field.

Using the observations made by Feldman *et al.* in [45], we show that our scheme is equivalent to the one proposed in the pioneering work of Cai and Yeung in [15]. However, with our approach, we can quickly and transparently recover some of the results available in the literature on secure network coding for wiretapped networks. The algorithm due to [15] is based on the code construction proposed by Li *et al.* in [72], however, more efficient network coding algorithms have been proposed recently (see, e.g., [13] and [73]). We use the results on the encoding complexity of the network coding presented in [73], [20], [21] to derive new bounds on the required field size of a secure network code, which are independent of the number of edges in the network, and which depend only on the number k of source symbols and the number t of destinations. We also propose an algorithm for constructing secure network codes achieving these bounds. Furthermore, we look at the dual problem and analyze the security of a given Ozarow-Wyner code, by studying the amount of information that can be gained by the wiretapper as a function of the number of wiretapped edges.

This chapter is organized as follows: In Section B, we briefly review the Ozarow-Wyner wiretap channel of type II problem. In Section C, we introduce the network generalization of this problem. In Section D, we present an algorithm for secure network code design and establish new bounds on the required code alphabet size. In Section E, we study the security performance of the Ozarow-Wyner codes. In Section F, we highlight some connections of this work with other works on secure network coding and network error correction.

B. Wiretap Channel II

We consider first a point-to-point scenario in which the source can transmit n symbols to the receiver, and an adversary can access any μ of those symbols [18, 19]. For this case, we know that the maximum number of symbols that the source can communicate to the receiver securely in the information-theoretic sense is equal to $n - \mu$.

The problem is mathematically formulated as follows. Let $S = (s_1, s_2, \dots, s_k)^T$ be the random variable associated with the k information symbols that the source wishes to send securely; $Y = (y_1, y_2, \dots, y_n)^T$, the random variable associated with the symbols that are transmitted through the noiseless channel between the source and the receiver, and $Z = (z_1, z_2, \dots, z_\mu)^T$ the random variable associated with the wiretapped symbols of Y . When $k \leq n - \mu$, there exists an encoding scheme that maps S into Y such that:

1. The uncertainty about S is not reduced by the knowledge of Z (perfect secrecy condition), i.e.,

$$H(S|Z) = H(S), \quad (3.1)$$

and,

2. The information S is completely determined by the complete knowledge of Y , that is,

$$H(S|Y) = 0. \quad (3.2)$$

For $n = 2$, $k = 1$ and $\mu = 1$, such coding scheme can be constructed as follows. If the source bit is 0, then either 00 or 11 is transmitted through the channel with equal probability. On the other hand, if the source bit is 1, then either 01 or 10 is transmitted through the channel with equal probability:

source bit s_1	0	1
codeword y_1y_2 chosen at random from	{00, 11}	{01, 10}

It is easy to see that knowledge of either y_1 or y_2 does not reduce the uncertainty about s_1 , whereas the knowledge of both y_1 and y_2 is sufficient to completely determine s_1 , namely, $s_1 = y_1 + y_2$.

In general, $k = n - \mu$ symbols can be transmitted securely by a coding scheme based on an $[n, n - k]$ linear maximal distance separable (MDS) code $\mathcal{C} \subset \mathbb{F}_q^n$. In this scheme, the encoder is a probabilistic device which operates on the space \mathbb{F}_q^n partitioned into q^k cosets of \mathcal{C} , where q is a large enough prime power. The k information symbols are taken as the syndrome which specifies a coset, and the transmitted word is chosen uniformly at random from the specified coset. The decoder recovers the information symbols simply by computing the syndrome of the received word. Because of the properties of MDS codes, knowledge of any $\mu = n - k$ or fewer symbols will leave the uncertainty of the k information symbols unchanged. For instance, the code used in the above example is the $[2, 1]$ repetition code with the following parity check matrix

$$\mathcal{H} = \begin{bmatrix} 1 & 1 \end{bmatrix}. \quad (3.3)$$

C. Wiretap Network II

We now consider an acyclic multicast network $G = (V, E)$ with unit capacity edges, an information source, t receivers, and the value of the min-cut to each receiver is equal to n . The goal is to maximize the multicast rate with the constraint of revealing no information about the multicast data to the wiretapper that can access data on any μ edges. We assume that the wiretapper knows the implemented network code, i.e., all

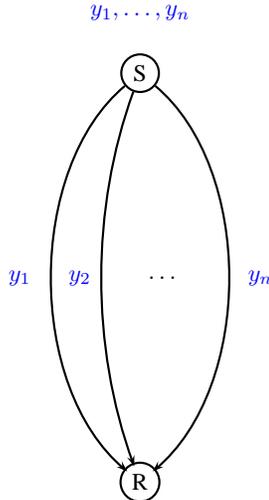


Fig. 21. Network equivalent to the wiretap channel of type II.

the coefficients of the linear combinations that determine the packets on each edge. Moreover, we assume that there is no shared randomness between the source and the receivers. The latter assumption rules out the use of traditional “key” cryptography to achieve security.

It can be seen that the wiretap channel of type II is equivalent to the simple unicast network of Figure 21 comprising n disjoint edges between the source and the destination, each carrying a different symbol. For this network, the source can multicast $k \leq n - \mu$ symbols securely by first applying a secure wiretap channel code (as described above) mapping k information symbols into n transmitted symbols (y_1, \dots, y_n) .

For general networks, when security is not an issue, a multicast rate n is possible with linear network coding [?, 72]. It is interesting to ask whether, using the same network code, the source can always multicast $k \leq n - \mu$ symbols securely using a wiretap channel code at the source. Naturally, this would be a solution if a multicast

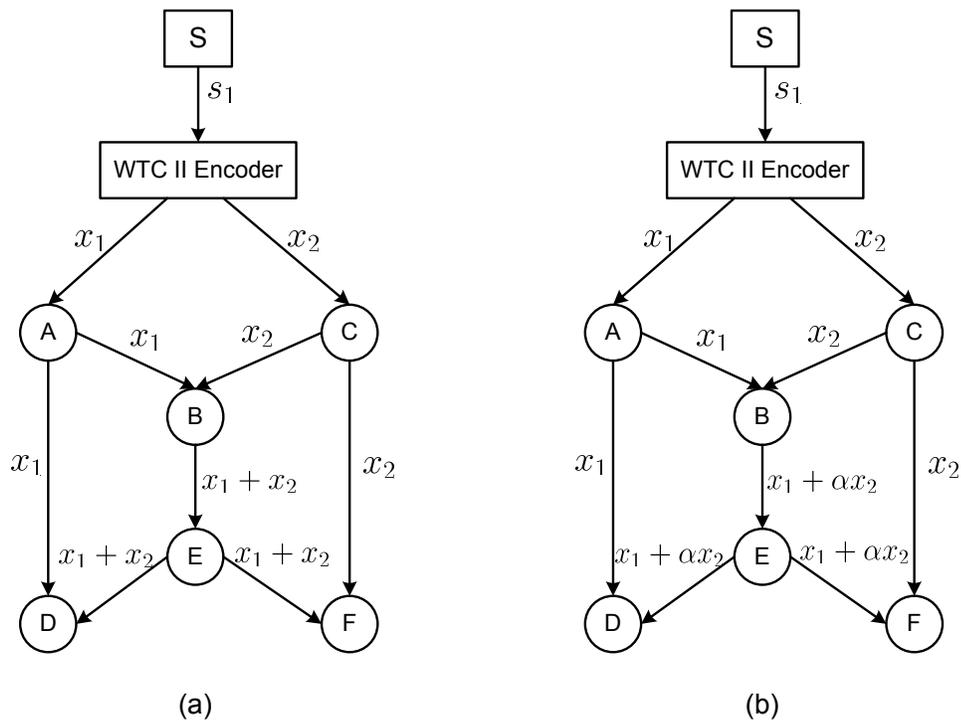


Fig. 22. Single-edge wiretap butterfly network with a) non-secure network code and b) secure network code. Security is achieved by using a coset encoder on top of the network.

rate of value n can be achieved just by routing.

Example C.1 (Butterfly Network). *Consider this approach for the butterfly network shown in Figure 22 where we have $n = 2$, $k = 1$ and $\mu = 1$. If the source applies the coding scheme described in the previous section and the usual network code as in Figure 22(a), the wiretapper will be able to learn the source symbol if it taps into any of the edges BE , EF or ED . Therefore, a network code can break down a secure wiretap channel code. However, if the network code is changed so that node B combines its inputs over, e.g., \mathbb{F}_3 and the coding vector of edge BE is $\begin{bmatrix} 1 & \alpha \end{bmatrix}$ where α is a primitive element of \mathbb{F}_3 (i.e., the message sent on edge BE is $x_1 + \alpha x_2$ as in Figure 22(b)), the wiretap channel code remains secure, that is, the adversary cannot gain any in-*

formation by accessing a single edge in the network. Note that the wiretap channel code based on the MDS code with $\mathcal{H} = \begin{bmatrix} 1 & 1 \end{bmatrix}$ remains secure with any network code whose BE coding vector is linearly independent of $\begin{bmatrix} 1 & 1 \end{bmatrix}$.

Next, we will show that the source can multicast $k \leq n - \mu$ symbols securely if it first applies a secure wiretap channel code based on an MDS code with a $k \times n$ parity check matrix \mathcal{H} if the network code is such that no linear combination of $\mu = n - k$ or fewer coding vectors belongs to the space spanned by the rows of \mathcal{H} . Let $W \subset E$ denote the set of $|W| = \mu$ edges the wiretapper chooses to observe, and $Z_W = (z_1, z_2, \dots, z_\mu)^T$ the random variable associated with the packets carried by the edges in W . Let C_W denote the matrix whose rows are the coding vectors associated with the observed edges in W . As in the case of the wiretap channel, let $S = (s_1, s_2, \dots, s_k)^T$ denotes the random variable associated with the k information symbols that the source wishes to send securely, and $Y = (y_1, y_2, \dots, y_n)^T$ the random variable associated with the n wiretap channel code symbols. The n symbols of Y will be multicast through the network using a linear network code. Writing $H(S, Y, Z_W)$ in two different forms, and taking into account the decodability condition of Equation (3.2), we get

$$H(S|Z_W) + H(Y|SZ_W) = H(Y|Z_W) + \underbrace{H(S|YZ_W)}_{=0}. \quad (3.4)$$

Our objective is to conceal all the information data from the wiretapper. The perfect secrecy condition implies

$$H(S|Z_W) = H(S), \forall W \subset E \text{ s.t. } |W| = \mu.$$

Thus, we obtain

$$H(Y|SZ_W) = H(Y|Z_W) - H(S). \quad (3.5)$$

This implies, in turn, that

$$n - \text{rank}(C_W) - k \geq 0. \quad (3.6)$$

Since there is a choice of edges such that $\text{rank}(C_W) = \mu$, the maximum rate for secure transmission is bounded as

$$k \leq n - \mu.$$

If the bound is achieved with equality, we have $H(Y|SZ_W) = 0$ and, consequently, the system of equations

$$\begin{bmatrix} S \\ Z_w \end{bmatrix} = \begin{bmatrix} \mathcal{H} \\ C_W \end{bmatrix} \cdot Y$$

has to have a *unique solution* for all W for which $\text{rank}(C_W) = \mu$. That is,

$$\text{rank} \begin{bmatrix} \mathcal{H} \\ C_W \end{bmatrix} = n \quad \text{for all } C_W \text{ s.t. } \text{rank}(C_W) = \mu. \quad (3.7)$$

This analysis proves the following result:

Theorem C.2. *Let $G = (V, E)$ be an acyclic multicast network with unit capacity edges and an information source such that the size of a minimum cut between the source and each receiver is equal to n . Then, a wiretap code at the source based on a coset code with a $k \times n$ parity check matrix \mathcal{H} and a network code such that no linear combination of $\mu = n - k$ or fewer coding vectors belongs to the space spanned by the rows of \mathcal{H} make the network information-theoretically secure against a wiretapper who can observe at most $\mu \leq n - k$ edges. Any adversary able to observe more than $n - k$ edges will have uncertainty about the source smaller than k .*

Next, we give an application of the previous theorem to the family of *combination* networks illustrated in Figure 23.

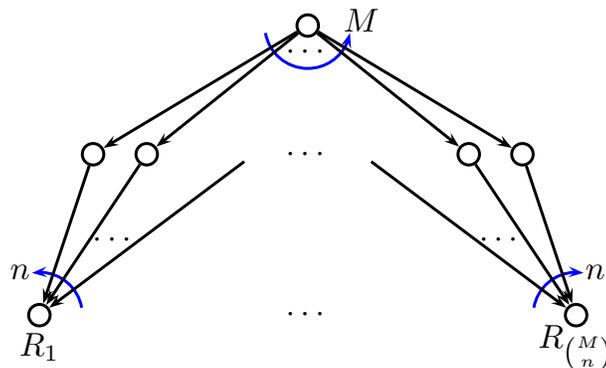


Fig. 23. The combination network $B(n, M)$.

Example C.3 (Combination Networks). A combination network $B(n, M)$ is defined over a 3-partite graph comprising three layers. The first layer contains a single source node, the second layer M intermediate nodes and the last layer is formed by $\binom{M}{n}$ receiver nodes such that every set of n nodes of the second layer is connected to a receiver.

The result of Theorem C.2 can be used to construct a secure network code for $B(n, M)$ using a $[M + k, M + k - n]$ MDS code which would achieve perfect secrecy against a wiretapper that can observe any $\mu = n - k$ edges. Let \mathcal{H} be an $n \times (M + k)$ parity check matrix of such MDS code over \mathbb{F}_q . A secure network code can be obtained by taking the first k rows of \mathcal{H}^T to form the matrix of the coset code at the source, and the rest of the rows of \mathcal{H}^T to be the coding vectors of the M edges going out of the source. Equation (3.7) is satisfied since the considered code is MDS and, therefore, any n columns of \mathcal{H} form a basis of \mathbb{F}_q^n . For instance if $M + k + 1$ is equal to a prime power q , a secure network code can be derived based on an $[M + k, M + k - n]$

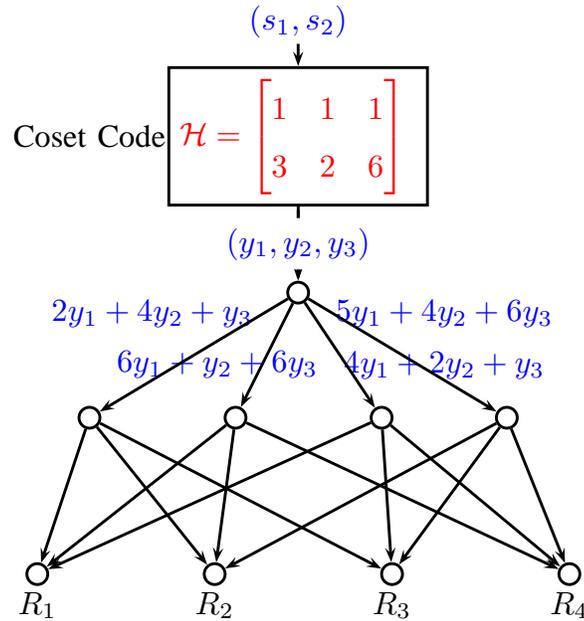


Fig. 24. A secure network code for the $B(3,4)$ combination network based on a $[6,3]$ Reed-Solomon code over \mathbb{F}_7 .

Reed-Solomon code with the following Vandermonde parity check matrix

$$\mathcal{H} = \begin{bmatrix} 1 & \alpha & \dots & \alpha^{M+k-1} \\ 1 & \alpha^2 & \dots & \alpha^{2(M+k-1)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^n & \dots & \alpha^{n(M+k-1)} \end{bmatrix}, \quad (3.8)$$

where α is a primitive element of \mathbb{F}_q . Figure C depicts a secure network code for the network $B(3,4)$ and $k = 2$ using a $[6,3]$ Reed-Solomon code over \mathbb{F}_7 whose parity check matrix is given by Equation (3.8) for $\alpha = 3$.

The above analysis shows that the maximum throughput can be achieved by applying a coset code at the source and then designing the network code while respecting certain constraints. The decoding of source symbols S is then merely a

matrix multiplication of the multicast symbols Y since $\mathcal{H}Y = S$. The method gives us a better insight on how much information the adversary can obtain if he can access more edges than the code is designed to combat. It also enables us to design secure network coding schemes over smaller alphabets. These two issues are discussed in detail in the next two sections.

D. Network Code Design and Alphabet Size

The approach described previously in the literature for finding a secure multicast network code consisted of decoupling the problem of designing the multicast network code from making it secure by using a coset code. Feldman *et al.* showed in [45] that there exist networks where the above construction requires a large field size. We present here a different construction that exploits the topology of the network by incorporating the security constraints into the *Linear Information Flow* (LIF) algorithm of Jaggi *et al.* [13] for constructing linear multicast network codes. As a result, we obtain an improved lower bound on the sufficient field size. However, the modified algorithm does not have polynomial time complexity like the original one.

We start by giving a brief high level overview of the LIF algorithm of [13]. The inputs of the algorithm are the network, the source node, the t receivers and the number n of information packets. Assuming the min-cut between the source and any receiver is at least n , the algorithm outputs a linear network code that guaranties the delivery of the n packets to all the receivers.

The algorithm starts by 1) finding t flows F_1, F_2, \dots, F_t of value n each, from the source to each receiver and 2) defining t $n \times n$ matrices B_{F_j} (one for each receiver) formed by the global encoding vectors of the n last visited edges in the flow F_j . Initially, each matrix B_{F_j} is equal to the identity matrix. Then, the algorithm loops

over all the network edges, visiting them in topological order. In each iteration, the algorithm finds a suitable local encoding vector for the visited edge, and updates accordingly all of the t matrices B_{F_j} . The algorithm maintains the invariant that the matrices B_{F_j} remain invertible after each iteration. Therefore, when it terminates, each receiver will get n linear combinations of the original packets that form a full rank system. Thus, each destination can solve for the original packets by inverting the corresponding matrix.

The analysis of the algorithm due to [13] implies that a field of size at least t (the number of destinations) is sufficient for finding the desired network code. In particular, as shown in [13, Lemma 8], a field of size larger or equal to t is sufficient for satisfying the condition that the t matrices B_{F_j} are always invertible.

To construct a secure network code, we modify the LIF algorithm in the following way. We select a $k \times n$ parity check matrix \mathcal{H} . Without loss of generality, we assume that the μ packets observed by the wiretapper are linearly independent, i.e., $\text{rank}C_W = \mu$. We denote by e_i the edge visited at the i -th iteration of the LIF algorithm, and by P_i the set of edges that have been processed in previous $i - 1$ iterations. Then, we extend the set of invariants to guaranty that the encoding vectors are chosen so that the matrices $M_W = \begin{bmatrix} \mathcal{H} \\ C_W \end{bmatrix}$ are also invertible; which, by Theorem C.2, achieves the security condition. More precisely, using the same technique as the original LIF algorithm, we make sure that by the end of the i -th iteration, the matrices B_{F_j} and the matrices M_{W_i} are invertible; where $W_i = \{e_i\} \cup W'$ and W' is any subset of P_i containing $\mu - 1 = n - k - 1$ edges. The total number of matrices that need to be kept invertible in this modified version of the LIF algorithm is at most $\binom{|E|-1}{\mu-1} + t$. Thus, similarly to [13, Lemma 8], we obtain the following improved bound on the alphabet size for secure multicast:

Theorem D.1. *Let $G = (V, E)$ be an acyclic network with unit capacity edges and an information source such that the min-cut value between the source and each of the t receivers is equal to n . A secure multicast at rate $k \leq n - \mu$ in the presence of a wiretapper who can observe at most $\mu \leq n$ edges is possible over the alphabet \mathbb{F}_q of size*

$$q \geq \binom{|E| - 1}{\mu - 1} + t. \quad (3.9)$$

The bound given by Equation (3.9) can be further improved by realizing, as was first done in [73], that not all edges in the network carry different linear combinations of the original packets. Langberg *et al.* showed in [20] that the number of *encoding edges* in a *minimal* acyclic multicast network is bounded by $2n^3t^2$. Encoding edges create new packets by combining the packets received over the incoming edges of their tail nodes. A minimal multicast network does not contain redundant edges, i.e., edges that can be removed from the network without violating its optimality. Reference [21] presents an efficient algorithm for constructing a minimal acyclic network \widehat{G} from the original one G . This work also shows that a feasible network code for a minimal network can be used for the original network, albeit slight modifications.

The main idea of our scheme is to find a secure network code for the minimal network \widehat{G} , and then use the procedure described in [21] to construct a network code for the original network G which will also be secure. Consider the problem of finding secure network codes for \widehat{G} . This problem will not change if the wiretapper is not allowed to wiretap the *forwarding edges*, i.e., the edges that just forward packets received by their tail nodes. Therefore, the set of edges that the wiretapper might have access to consists of the encoding edges and the edges outgoing from the source. The number of such edges is bounded by $2n^3t^2$. Now, applying Theorem D.1 on \widehat{G} and

taking into consideration the restriction on the edges that can be potentially wire-tapped, we obtain the following bound on the sufficient field size which is independent of the size of the network.

Corollary D.2. *For the communication scenario of Theorem D.1, a secure linear multicast network code always exists over the alphabet \mathbb{F}_q of size*

$$q \geq \binom{2k^3t^2}{\mu - 1} + t. \quad (3.10)$$

For networks with two sources, we can completely settle the question on the required alphabet size for a secure network code. Note that the adversary has to be limited to observing at most one edge of his choice. Based on the work of Fragouli and Soljanin [73], we know that the coding problem for these networks is equivalent to a vertex coloring problem of some specially designed graphs, where the colors correspond to the points on the projective line $\mathbb{PG}(1, q)$:

$$[0 \ 1], [1 \ 0], \text{ and } [1 \ \alpha^i] \text{ for } 0 \leq i \leq q - 2, \quad (3.11)$$

where α is a primitive element of \mathbb{F}_q . Clearly, any network with two sources and any arbitrary number of receivers can be securely coded by reducing the set of available colors in (3.11) by removing one point (color) $[1 \ 1]$, and applying a wiretap code based on the matrix $\mathcal{H} = [1 \ 1]$ as in the example above. Alphabet size sufficient to find secure codes for all networks with two sources also follows from [73]:

Theorem D.3. *For any configuration with two sources and t receivers, the code alphabet \mathbb{F}_q of size*

$$\lfloor \sqrt{2t - 7/4} + 1/2 \rfloor + 1$$

is sufficient for a secure network code. There exist configurations for which it is necessary.

E. Wiretapper Equivocation

In this section, we analyze the performance of coset codes in the case of a wiretapper with variable strength, i.e., the number μ of edges he can observe is not fixed. For a given coset code, we seek to quantify the amount of information that is leaked to the wiretapper as a function of μ .

Assume that at the source s of a multicast network a coset code defined by a $k \times n$ parity check matrix \mathcal{H} is used as described in the previous section. The equivocation $\Delta(\mu)$ of the wiretapper, i.e., the uncertainty it has about the information source vector $S = (s_1, \dots, s_k)^T$, is defined, as in [19], based on the worst case scenario, by

$$\Delta(\mu) := \min_{W \subseteq E; |W|=\mu} H(S|Z_W), \quad (3.12)$$

where $Z_W = (z_1, \dots, z_\mu)^T$ is the random variable representing the observed packets on the set $W \subseteq E$ of wiretapped edges. We have $Z_W = C_W Y$ where C_W is an $\mu \times n$ matrix, and $Y = (y_1, \dots, y_n)^T$ is the output of the coset code at the source. It can be seen that $\Delta(\mu)$ can be written as:

$$\Delta(\mu) = \min_{\substack{W \subseteq E; |W|=\mu \\ \text{rank}(C_W)=\mu}} H(S|Z_W). \quad (3.13)$$

Therefore, we will assume from now on, without loss of generality, that W is such that $\text{rank}(C_W) = \mu$. For a given choice of such W , let C_W^\perp be the parity check matrix of the $[n, \mu]$ code generated by C_W . Let I_n be the $n \times n$ identity matrix. Define $J_{n,\mu}$ to be the $n \times (n - \mu)$ matrix where the first μ rows are all zeros, and the last $n - \mu$ rows form $I_{n-\mu}$. Theorem E.1 below gives the expression of $\Delta(\mu)$ which depends on the network code and the coset code used.

Theorem E.1.

$$\Delta(\mu) = \min_{\substack{W \subset E; |W|=\mu \\ \text{rank}(C_W)=\mu}} \text{rank} \left(\mathcal{H} \begin{bmatrix} C_W \\ C_W^\perp \end{bmatrix}^{-1} J_{n,\mu} \right). \quad (3.14)$$

Proof. First, let $A_W = \begin{bmatrix} C_W \\ C_W^\perp \end{bmatrix}$. By Equation (3.4), we have

$$\begin{aligned} H(S|Z_W) &= H(Y|Z_W) - H(Y|SZ_W) \\ &= n - \text{rank}(C_W) - \left(n - \text{rank} \begin{bmatrix} \mathcal{H} \\ C_W \end{bmatrix} \right) \\ &= \text{rank} \left(\begin{bmatrix} \mathcal{H} \\ C_W \end{bmatrix} A_W^{-1} \right) - \text{rank}(C_W) \\ &= \text{rank} \left(\begin{bmatrix} \mathcal{H} A_W^{-1} \\ C_W A_W^{-1} \end{bmatrix} \right) - \text{rank}(C_W) \\ &= \dim(\langle \mathcal{H} A_W^{-1} \rangle) + \dim(\langle C_W A_W^{-1} \rangle) \\ &\quad - \dim(\langle \mathcal{H} A_W^{-1} \rangle \cap \langle C_W A_W^{-1} \rangle) - \text{rank}(C_W) \\ &= k - \dim(\langle \mathcal{H} A_W^{-1} \rangle \cap \langle J'_{n,\mu} \rangle), \end{aligned} \quad (3.15)$$

where $\langle \cdot \rangle$ denotes the row space of a matrix, and $J'_{n,\mu}$ is the $\mu \times n$ matrix where the first μ columns form I_μ and the last $n - \mu$ columns are all zeros. Note that $\dim(\langle \mathcal{H} A_W^{-1} \rangle \cap \langle J'_{n,\mu} \rangle)$ is exactly k minus the rank of the last $n - \mu$ column vectors of $\mathcal{H} A_W^{-1}$. \square

A relevant concept to our work here is that of the generalized Hamming weights $d_1(\mathcal{C}), \dots, d_k(\mathcal{C})$ of a linear code \mathcal{C} introduced by Wei in [74], and that characterize the performance of coset codes over the classical wiretap channel of Type II. The generalized Hamming weights were extended to the wiretap networks setting in [75].

Given a certain network with an associated network code and coset code, Theorem E.1 provides an equivalent expression of the network formulation of the r -th generalized Hamming weight d_r as the minimum number of edges that should be wiretapped to leak r symbols to the wiretapper. Then, we can write

$$\begin{aligned} d_r &:= \min\{\mu; \Delta(\mu) = k - r\} \\ &:= \min\{\mu; \min_{\substack{W \subset E; |W|=\mu \\ \text{rank}(C_W)=\mu}} \text{rank} \left(\mathcal{H} \begin{bmatrix} C_W \\ C_W^\perp \end{bmatrix}^{-1} J_{n,\mu} \right) = k - r\}. \end{aligned} \quad (3.16)$$

Next, we focus on three special cases. First, we revisit the model of the wiretap channel of type II of [18]. Second, we consider the case where the wiretapper may gain access to more edges than what the secure code is designed to combat. Third, we study the scenario where only a part of the network edges are vulnerable to wiretapping.

1. Wiretap Channel of Type II

Consider, again, the wiretap channel of type II studied in [18]. Theorem E.1 can be used to easily recover the following classical result.

Corollary E.2. *The equivocation rate of the wiretapper in the wiretap channel of type II is given by*

$$\Delta(\mu) = \min_{\substack{U \subseteq \{1,2,\dots,n\} \\ |U|=n-\mu}} \text{rank}\{\mathcal{H}_i; i \in U\}, \quad (3.17)$$

where \mathcal{H}_i denote the i th column of the parity check matrix \mathcal{H} .

Proof. The wiretap channel of type II is equivalent to the network depicted in Figure 21. Assume that the edges between the source and the destination are indexed from 1 to n , so that $E = \{1, \dots, n\}$. For any $W \subseteq \{1, \dots, n\}$, define I_W to be the

matrix formed by the rows of the $n \times n$ identity matrix indexed by the elements of W in an increasing order. Since edge i carries the packet y_i , for a given set $W \subseteq E$ of wiretapped edges, $C_W = I_W$ and $C_W^\perp = I_U$, where $U = \{1, \dots, n\} \setminus W$. Therefore, $A_W^{-1} = \begin{bmatrix} I_W \\ I_U \end{bmatrix}^{-1} = A_W^T$, and the last $n - \mu$ columns of $\mathcal{H}A_W^T$ are exactly the columns of \mathcal{H} indexed by U . \square

2. Underestimated Wiretapper

Suppose the coset code defined by the $k \times n$ parity check matrix \mathcal{H} satisfies Theorem C.2 and achieves perfect secrecy against a wiretapper that can observe λ edges. If, however, the wiretapper can access μ edges, where $\mu > \lambda$, then the amount of information leaked to the wiretapper can be shown to be equal to $\mu - \lambda$, i.e., the number of additional wiretapped edges.

Corollary E.3. *For the case of an underestimated wiretapper, the equivocation of the wiretapper is given by:*

$$\Delta(\mu) = k - (\mu - \lambda).$$

Proof. Since the coset code achieves perfect secrecy for λ wiretapped edges, by Theorem C.2, we have $k = n - \lambda$ and $H(S|YZ_W) = 0$. Thus, Equation (3.4) gives

$$H(S|Z_W) = H(Y|Z_W) = n - \text{rank}(C_W) = k + \lambda - \text{rank}(C_W).$$

The minimum value of $H(S|Z_W)$ is obtained when C_W has maximal rank, i.e., when $\text{rank}(C_W) = \mu$. \square

3. Restricted Wiretapper

In practice, for instance in large networks, the wiretapper may not have access to all the network edges, and his choice of μ edges is limited to a certain edge subset

$E' \subset E$. For this model, the equivocation rate of the wiretapper is determined by Equation (3.14) where E is replaced by E' . An interesting case arises, however, when the edges in E' belong to a cut of n edges between the source and one of the receivers. In this case, the performance of the coset code is the same as when it is used for a wiretap channel of type II.

Corollary E.4. *In the case of a restricted wiretapper that can observe any μ edges in a cut between the source and one of the destinations, the equivocation rate of the wiretapper is given by Equation (3.17).*

Proof. Assume the edges that are vulnerable to wiretapping are indexed from 1 to n , so that $E' = \{1, \dots, n\}$. Let $Z_{E'} = (z_1, \dots, z_n)^T$ denote the packets carried by those edges, such that edge i carries packet z_i . We can write $Z_{E'} = C_{E'}Y$, where $C_{E'}$ is an $n \times n$ matrix. Since the cut has n edges, the matrix $C_{E'}$ is invertible; otherwise, by the properties of linear network codes, the destination corresponding to the considered cut cannot decode Y . For a choice $W \subseteq E'$ of wiretapped edges, we have $Z_W = C_WY$, where $C_W = I_W C_{E'}$. Moreover, $C_W^\perp = I_{\bar{W}} C_{E'}$, where $\bar{W} = E' \setminus W$. Therefore,

$$\mathcal{H} \begin{bmatrix} C_W \\ C_W^\perp \end{bmatrix}^{-1} = \mathcal{H} \left(C_{E'} \begin{bmatrix} I_W \\ I_{\bar{W}} \end{bmatrix} \right)^{-1} = \mathcal{H} C_{E'}^{-1} \begin{bmatrix} I_W \\ I_{\bar{W}} \end{bmatrix}^T.$$

Similar to the proof of Corollary E.2, the last $n - \mu$ columns of $\mathcal{H}A^{-1} \begin{bmatrix} I_W \\ I_{\bar{W}} \end{bmatrix}^T$ are exactly the columns of $\mathcal{H}A^{-1}$ indexed by U . So, by Theorem E.1, we have

$$\begin{aligned} \Delta(\mu) &= \min_{\substack{U \subseteq \{1, 2, \dots, n\} \\ |U| = n - \mu}} \text{rank}\{(\mathcal{H}A^{-1})_i; i \in U\} \\ &= \min_{\substack{U \subseteq \{1, 2, \dots, n\} \\ |U| = n - \mu}} \text{rank}\{\mathcal{H}_i; i \in U\}. \end{aligned}$$

□

Note that the previous result still holds for any subset E' of possible wire-tapped edges such that $C_{E'}$ is invertible. For this scenario, the equivocation rate of the wiretapper can be alternatively given by the generalized Hamming weights [74] $d_1(\mathcal{C}), \dots, d_k(\mathcal{C})$ of the linear code \mathcal{C} generated by \mathcal{H} . In this case, for a given μ , $\Delta(\mu)$ is the unique solution to the following inequalities [74, Cor. A]:

$$d_{n-\mu-\Delta(\mu)}(\mathcal{C}) \leq n - \mu < d_{n-\mu-\Delta(\mu)+1}(\mathcal{C}).$$

F. Connections with Other Schemes

In this section, we explore the relationship between our proposed scheme and previously known constructions.

1. Secure Network Coding and Filtered Secret Sharing

Cai and Yeung were first to study the design of secure network codes for multicast demands [15]. They showed that, in the setting described above, a secure network code can be found for any $k \leq n - \mu$. Their construction is equivalent to the following scheme:

1. Generate a vector $R = (r_1, r_2, \dots, r_\mu)^T$ choosing its components uniformly at random over \mathbb{F}_q ,
2. Form vector X by concatenating the μ random symbols R to the k source symbols S :

$$X = \begin{bmatrix} S \\ R \end{bmatrix} = (s_1, \dots, s_k, r_1, \dots, r_\mu)^T$$

3. Choose an *invertible* $n \times n$ matrix T over \mathbb{F}_q and a feasible multicast network code [72] to ensure the security condition given by Equations (3.1) and (3.2)¹.
4. Compute $Y = TX$, and multicast Y to all the destinations by using the constructed code.

Feldman *et al.* considered also the same problem in [45]. Adopting the same approach of [15], they showed that in order for the code to be secure, the matrix T should satisfy certain conditions ([45, Thm. 6]). In particular, they showed that in the above transmission scheme, the security condition of Equations (3.1) and (3.2) holds if and only if any set of vectors consisting of

1. at most μ linearly independent edge coding vectors, and
2. any number of vectors from the first k rows of T^{-1}

is linearly independent. They also showed that if one sacrifices in the number of information packets, that is, take $k < n - \mu$, then it is possible to find secure network codes over fields of size much smaller than the very large bound $q > \binom{|E|}{\mu}$.

We will show now that our approach based on the coset codes for the wiretap channel at the source is equivalent to the scheme of [15] with the conditions of [45].

Proposition F.1. *For any $n \times n$ matrix T satisfying the security conditions defined above, the $k \times n$ matrix $\mathcal{H} = T^*$ formed by taking the first k rows of T^{-1} satisfies the condition of Theorem C.2.*

Proof. Consider the secure multicast scheme of [15] as presented above. For a given information vector $S \in \mathbb{F}_q^k$, let $B(S)$ be the set of all possible vectors $Y \in \mathbb{F}_q^n$ that

¹It is shown in [15, Thm. 1] that such code and matrix T exist provided that $q > \binom{|E|}{\mu}$.

could be multicast through the network under this scheme. More precisely,

$$B(S) = \left\{ Y \in \mathbb{F}_q^n \mid Y = TX, X = \begin{bmatrix} S \\ R \end{bmatrix}, R \in \mathbb{F}_q^{n-k} \right\}.$$

Then, for all $Y \in B(S)$, we have $T^*Y = T^*T \begin{bmatrix} S \\ T \end{bmatrix} = S$. Therefore, any $Y \in B(S)$ also belongs to the coset of the space spanned by the rows of T^* whose syndrome is equal to S . Moreover, since T is invertible, $|B(S)| = 2^{n-k}$ implies that the set $B(S)$ is exactly that coset. The conditions of [45] as stated above translate directly into Equation (3.7), the remaining condition of Theorem C.2. \square

2. Universal Secure Network Codes

In practical implementations of linear multicast network codes over \mathbb{F}_q , the information sources are typically packets of a certain length m , i.e., s_1, \dots, s_k are vectors in \mathbb{F}_q^m . Applying our approach here which appeared in a preliminary version in [16], Silva and Kschischang devised in [56] a scheme that achieves a complete decoupling between the secure code and the network code design. Their scheme is universal in the sense that it achieves secrecy by applying a coset code at the source with no knowledge of the network code used. The main idea is to use a special class of MDS codes called maximal rank-distance codes (MRD) which are non-linear over \mathbb{F}_q , but linear over the extension field \mathbb{F}_{q^m} . The parity check matrix of an MRD code over \mathbb{F}_{q^m} has the interesting property that it always satisfies the condition of Theorem C.2 when the edge coding vectors are over \mathbb{F}_q , as stated in the theorem below.

Lemma F.2. [56, Lemma 3] *Let \mathcal{H} be the parity check matrix of an $[n, n-k]$ linear MRD code over \mathbb{F}_{q^m} . For any full rank $(n-k) \times n$ matrix B over \mathbb{F}_q , the $n \times n$ matrix $\begin{bmatrix} \mathcal{H} \\ B \end{bmatrix}$ is invertible.*

Therefore, MRD codes will always achieve perfect secrecy irrespective of the network code used. The choice of the MRD code will only depend on the underlying field \mathbb{F}_q of the network code used.

3. Byzantine Adversaries

The malicious activity of the wiretapper in the model considered here was restricted to eavesdropping. A more powerful wiretapper, with jamming capabilities, may not only listen to the data in the network but also alter it. This may lead to flooding the whole network with erroneous packets. Schemes to combat such wiretappers, known in literature as Byzantine adversaries, were studied in [52, 36, 55], and the references within.

Consider a scenario where the wiretapper can, not only observe μ edges, but also jam α edges of his choice that are unknown to the destinations. In this case, we will describe a coding scheme that achieves a multicast rate of $k = n - 2\alpha - \mu$ and guarantees that the information will remain hidden from the wiretapper. This can be achieved by using a coset code as described in Section C, followed by a powerful network error-correcting code [53, 54]. First, we recall an important result from [54, Theorem 4].

Theorem F.3. *For an acyclic network $G(V, E)$ with min-cut n , there exists a linear α -error-correcting code of dimension $(n - 2\alpha)$ over a sufficiently large field.*

Let \mathcal{G} be the generator matrix of a linear α -error-correcting code of dimension $(n - 2\alpha)$ whose existence is guaranteed by the previous theorem, and Let \mathcal{G}^\perp be its parity check matrix. A block diagram of the coding scheme that achieves secrecy against a Byzantine wiretapper at a rate $k = n - 2\alpha - \mu$ is depicted in Figure 25. First, the information $S = (s_1, \dots, s_k)^T$ is encoded using a coset code of parity check

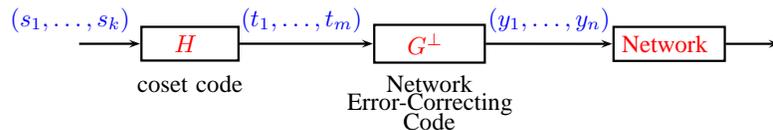


Fig. 25. A coding scheme achieving perfect secrecy against a limited Byzantine wire-tapper.

matrix \mathcal{H} into the vector $T = (t_1, \dots, t_m)^T$, with $m = k + \mu$. The vector T is then encoded into $Y = (y_1, \dots, y_n)^T = \mathcal{G}T$ using the network error-correcting code. To achieve perfect secrecy, \mathcal{H} should satisfy the condition of Theorem C.2, which can be expressed here as:

$$\text{rank} \begin{bmatrix} \mathcal{H} \\ C_W \mathcal{G} \end{bmatrix} = k + \mu, \quad \text{for all } C_W \text{ s.t. } \text{rank}(C_W) = \mu. \quad (3.18)$$

We assume that the code is over a field large enough to guaranty the existence of the network error-correcting code and the existence of the matrix \mathcal{H} satisfying the above condition as well. At each destination, a decoder corrects the errors introduced by the wiretapper and recovers T . The information S is then obtained as the unique solution of the system $\mathcal{H}S = T$. It was shown in [76] that the rate $k = n - 2\alpha - \mu$ is optimal and another construction for codes with the same properties was presented there.

CHAPTER IV

NETWORK CODING, INDEX CODING AND MATROID THEORY

We analyze in this chapter the relation between index coding, network coding, and matroid linear representation. We devise a reduction from the index coding problem to the network coding problem, implying that in the linear case these two problems are equivalent. We also present a second reduction from the matroid linear representability problem to index coding, and therefore to network coding. The latter reduction establishes a strong connection between matroid theory and network coding theory. These two reductions are then used to construct special instances of the index coding problem where vector codes outperform scalar linear ones, and where non-linear encoding is needed to achieve the optimum number of transmissions. Thereby, we provide a counterexample to a related conjecture in the literature and answer a question on the benefits of vector linear codes. The results presented in this chapter have appeared in [23, 24, 25]¹.

A. Introduction

In recent years, there has been a significant interest in utilizing the broadcast nature of wireless signals for improving the throughput and reliability of ad-hoc wireless networks. The wireless medium allows the transmitter to deliver data to several neighboring nodes with a single transmission. Moreover, a wireless receiver can op-

¹Parts of this chapter are reprinted with permission from “On the Relation Between the Index Coding and the Network Coding Problems,” by S. El Rouayheb, A. Sprintson, and C. Georghiades, In proceedings of 2008 IEEE International Symposium on Information Theory (ISIT), Toronto, Canada, July, 2008, pages 1823- 1827 and “A New Construction Method for Networks From Matroids,” by S. El Rouayheb, A. Sprintson and C. N. Georghiades, 2009 IEEE International Symposium on Information Theory (ISIT), Seoul, Korea, June, 2009, pages 2872-2876, copyright IEEE.

portunistically listen to the wireless channel and store all the overheard packets, including those designated for different users. As a result, wireless nodes can obtain side information which, in combination with proper encoding techniques, can lead to a substantial improvement in the performance of the wireless network.

Several recent studies focused on wireless architectures that use coding techniques to take advantage of the inherent properties of wireless communications. In particular, [77, 78] proposed new architectures, referred to as COPE and MIXIT, in which routers mix packets from different information sources to increase the overall network throughput. Birk and Kol [11, 79] discussed applications of coding techniques in satellite networks with caching clients with a low-capacity reverse channel [11, 79].

The major challenge in the design of opportunistic wireless networks is to identify an optimal encoding scheme that minimizes the number of transmissions necessary to satisfy all the receiver nodes. This can be formulated as the *index coding* problem [10] that includes a single transmitter node s and a set of receiver nodes R . The sender has a set of information messages $X = \{x_1, \dots, x_k\}$ that need to be delivered to the receiver nodes. Each receiver $\rho = (x, H) \in R$ needs to obtain a single message $x \in X$ and has prior *side information* in the form of a subset H of X . The sender can broadcast the encoding of messages in X to the receivers through a noiseless channel that has a capacity of one message per channel use. The objective is to find an optimal encoding scheme, referred to as an *index code*, that satisfies all the receiver nodes with a minimum number of transmissions.

With *linear coding*, all messages in X are taken to be elements of a finite field, and all encoding operations are linear over that field. Figure 26 depicts an instance of the index coding problem that includes a transmitter with four messages x_1, \dots, x_4 and

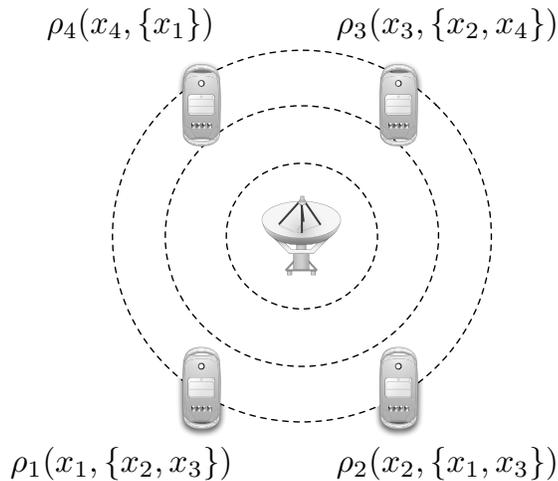


Fig. 26. An instance of the index coding problem with four messages and four receivers ρ_1, \dots, ρ_4 . Each receiver ρ_i is represented by a couple (x, H) , where $x \in X$ is the packet demanded by the receiver, and $H \subseteq X$ represent its side information.

four receivers. We assume in this example that each message is an element of $GF(2^n)$, represented by n bits. Note that the sender can satisfy the demands of all clients in a straightforward way by using four transmissions to broadcast all the messages over the wireless channel. The encoding operation allows to reduce the number of messages by a factor of two. Indeed, it is sufficient to only send the two messages $x_1 + x_2 + x_3$ and $x_1 + x_4$ (all operations are in $GF(2^n)$) to satisfy the requests of all clients. This example demonstrates that by using an efficient code, the sender can significantly reduce the number of transmissions which, in turn, results in reduction in delay and energy consumption.

The above example utilizes a *scalar linear* encoding scheme that performs coding over the original messages. In a *vector* encoding scheme, each message is divided into a number of smaller size messages, referred to as *packets*. The vector encoding scheme

combines packets from different messages to minimize the number of transmissions. With *vector linear* index coding, all packets are elements of a certain finite field \mathbb{F} , and each transmitted packet is a linear combination of the original packets. For example, consider the instance depicted in Figure 26, and suppose that each message x_i is divided into two packets, $x_i^1, x_i^2 \in GF(2^{n-1})$. Then, a valid vector-linear solution is comprised of four packets $\{x_1^1 + x_4^1, x_1^2 + x_4^2, x_1^1 + x_2^1 + x_3^1, x_1^2 + x_2^2 + x_3^2\}$.

In this chapter, we study the relation between the index coding problem and the more general network coding problem. In particular, we establish a reduction that maps any instance of the network coding problem to a corresponding instance of the index coding problem. We show that several important properties of the Network coding problem carry over to index coding. Specifically, by applying our reduction to the network constructed in [5], we show that vector linear index codes are suboptimal. We also present an instance of the index coding problem in which splitting a message into two packets yields a smaller number of transmissions than a scalar linear solution.

We also study the relation between index coding and matroid theory. In particular, we present a reduction that maps any matroid to an instance of the index coding problem such that the problem has a “special” optimal vector linear code if and only if the matroid has a multilinear representation. Using results on the non-Pappus matroid, we give another example where vector linear index codes outperform scalar linear ones. Moreover, this reduction establishes a strong connection between network coding and matroid theory and constitutes a means to apply numerous results in the rich field of matroid theory to communication problems in networks.

The rest of this chapter is organized as follows. In Section B, we discuss our model and give a formulation of the index and network coding problems. In Section C, we present a reduction from the network coding problem to the index coding problem that shows that these two problems are equivalent for the case of linear codes. In

Section D, we present our second reduction that establishes the relation between index codes and matroid linear representation. In Section E, we discuss applications of our reductions to derive results on the optimality of linear codes. In Section F, we discuss the connection between matroids and networks.

B. Model

In this section, we give the mathematical model for the index coding problem. In addition, we present a formulation of the network coding problem that is equivalent to the one described in the previous chapters but that is more suitable to the study here.

1. Index Coding

An instance of the index coding problem $\mathcal{I}(X, R)$ includes

1. A set of k messages $X = \{x_1, \dots, x_k\}$,
2. A set of clients or receivers $R \subseteq \{(x, H); x \in X, H \subseteq X \setminus \{x\}\}$.

Here, X represents the set of messages available at the transmitter. Each message x_i can be divided into n packets each belonging to a finite alphabet Σ , and we write $x = (x_{i1}, \dots, x_{in}) \in \Sigma^n$. We denote by $\xi = (x_{11}, \dots, x_{1n}, \dots, x_{k1}, \dots, x_{kn}) \in \Sigma^{nk}$ the concatenation of all the packets available at the transmitter.

A receiver is represented by a pair (x, H) , where $x \in X$ is the message it demands and $H \subseteq X \setminus \{x\}$ is the set of messages representing its side information. Note that in this model each receiver requests exactly one message. This does not incur any loss of generality since any receiver requesting multiple messages can be substituted by several receivers that demand a single different message and have the same side information as the original one.

Definition B.1 (Index Codes). An (n, q) index code for $\mathcal{I}(X, R)$ is a function $f : \Sigma^{nk} \rightarrow \Sigma^c$, for a certain positive integer c , such that for each client $\rho = (x, H) \in R$, there exists a function $\psi_\rho : \Sigma^{c+n|H|} \rightarrow \Sigma^n$ such that $\psi_\rho(f(\xi), (x_i)_{x_i \in H}) = x, \forall \xi \in \Sigma^{nk}$.

We refer to c as the *length* of the index code and c/n as its *normalized length*. Since the wireless broadcast channel has a capacity of one message per transmission, the normalized length of the index code represents the corresponding number of transmissions. Define $\ell(n, q)$ to be the smallest integer c such that the above condition holds for the given alphabet size q and block length n . If the index code satisfies $c = \ell(n, q)$, it is said to be *optimal*. Given n and q , the *index coding problem* consists of finding an optimal (n, q) index code for an index coding instance. We denote by $\lambda(n, q) = \ell(n, q)/n$ the normalized length, i.e., minimum number of transmissions, for an (n, q) index code solution of a given instance $\mathcal{I}(X, R)$ of the index coding problem. We are also particularly interested in the minimum number of transmissions that can be achieved by (n, q) linear codes, and we denote it by $\lambda^*(n, q)$.

We refer to ψ_ρ as the *decoding function* for client ρ . In the case of a linear index code, the alphabet Σ is a field and the functions f and ψ_ρ are linear in the variables x_{ij} . In this case, if $n = 1$ the index code is called a scalar linear code, and for $n > 1$, it is called a vector linear index code of dimension n . Note that in our formulation model, a message can be requested by several clients. This is a slightly more general model than the one considered in references [10] and [12] where it was assumed that each message can only be requested by a single client.

Let $\mu(\mathcal{I})$ be the largest number of messages requested by a set of clients with identical side information, i.e., $\mu(\mathcal{I}) = \max_{Y \subseteq X} |\{x_i; (x_i, Y) \in R\}|$. Then, it is easy to verify that the minimum number of transmissions $\lambda(n, q)$ is lower-bounded by $\mu(\mathcal{I})$, independently of the values of n and q . To see this, let $Y^* = \arg \max_{Y \subseteq X} |\{x_i; (x_i, Y) \in R\}|$

and $W = \{x_i; (x_i, Y^*) \in R\}$ and remove all clients that do not have the set Y^* as side information. We note that, since $Y^* \cap W = \emptyset$, the minimum number of transmissions corresponding to the resulting instance is equal to $|W| = \mu(\mathcal{I})$ and since it cannot be larger than $\lambda(n, q)$, we get $\lambda(n, q) \geq \mu(\mathcal{I})$.

2. Network Coding

Let $G(V, E)$ be a directed acyclic graph with vertex set V and edge set E . For each edge $e(u, v) \in E$, we define the in-degree of e to be the in-degree of its tail node u , and its out-degree to be the out-degree of its head node v . Furthermore, we define $\mathcal{P}(e)$ to be the set of the parent edges of e , i.e., $\mathcal{P}(e(u, v)) = \{(w, u); (w, u) \in E\}$. Let $S \subset E$ be the subset of the edges in E of zero in-degree, and let $D \subset E$ be the subset of the edges of zero out-degree. We refer to the edges in S as *input* edges, and those in D as *output* edges. Also, we define $m = |E|$ to be the total number of edges, $k = |S|$ be the total number of input edges in the graph G , and $d = |D|$ be the total number of output edges. Moreover, we assume that the edges in E are indexed from 1 to m such that $E = \{e_1, \dots, e_m\}$, $S = \{e_1, \dots, e_k\}$ and $D = \{e_{m-d+1}, \dots, e_m\}$.

We model a communication network by a pair $\mathcal{N}(G(V, E), \delta)$ formed by a graph $G(V, E)$ and an onto function $\delta : D \rightarrow S$ from the set of output edges to the set of input edges. We assume that the tail node of each input edge e_i , $i = 1, \dots, k$ holds the message x_i , also denoted as $x(e_i)$. The edges of the graph represent communication links that can transmit one message. The function $\delta(\cdot)$ specifies for each output edge e_i , $i = m - d + 1, \dots, m$, the source message $x(\delta(e_i))$ demanded by its head node. We refer to $\delta(\cdot)$ as the *demand function*. Each message x_i can be divided into n packets, each belonging to a finite alphabet Σ ; and we write $x_i = (x_{i1}, \dots, x_{in}) \in \Sigma^n$. We also denote by $\xi = (x_{11}, \dots, x_{1n}, \dots, x_{k1}, \dots, x_{kn}) \in \Sigma^{nk}$ the concatenation of all the packets available at the input edges.

Definition B.2 (Network Code). A q -ary *network code* of block length n , or an (n, q) network code, for the network $\mathcal{N}(G(V, E), \delta)$ is a collection

$$\mathcal{C} = \{f_e = (f_e^1, \dots, f_e^n); e \in E, f_e^i : \Sigma^{nk} \longrightarrow \Sigma, i = 1, \dots, n\},$$

of functions, called *global encoding* functions, indexed by the edges of G , that satisfy, for all $\xi \in \Sigma^{nk}$, the following conditions:

(N1) $f_{e_i}(\xi) = x_i$, for $i = 1, \dots, k$;

(N2) $f_{e_i}(\xi) = x(\delta(e_i))$, for $i = m - d + 1, \dots, m$;

(N3) For each $e = (u, v) \in E \setminus S$ with $\mathcal{P}(e) = \{e_1, \dots, e_{p_e}\}$, there exists a function $\phi_e : \Sigma^{np_e} \longrightarrow \Sigma^n$, referred to as the *local encoding function* of e , such that $f_e(\xi) = \phi_e(f_{e_1}(\xi), \dots, f_{e_{p_e}}(\xi))$, where p_e is the in-degree of e , and $\mathcal{P}(e)$ is the set of parent edges of e .

We are mostly interested here in linear network codes where Σ is a finite field \mathbb{F} , and all the global and local encoding functions are linear functions of the packets x_{ij} . In this case, when $n = 1$, the network code is referred to as a *scalar* linear network code, otherwise, it is called a *vector* linear network code of dimension n . Note that, a scalar linear network code over $GF(q)$ will naturally induce a vector linear network code of block length n over $GF(q^n)$; however, the converse is not necessarily true.

C. Connection to Network Coding

Index coding can be regarded as a special case of network coding. Indeed, for every instance of the index coding problem and a given integer c , there exists an equivalent instance of the network coding problem that has an (n, q) network code solution if and only if there exists an (n, q) index code of normalized length c . For example,

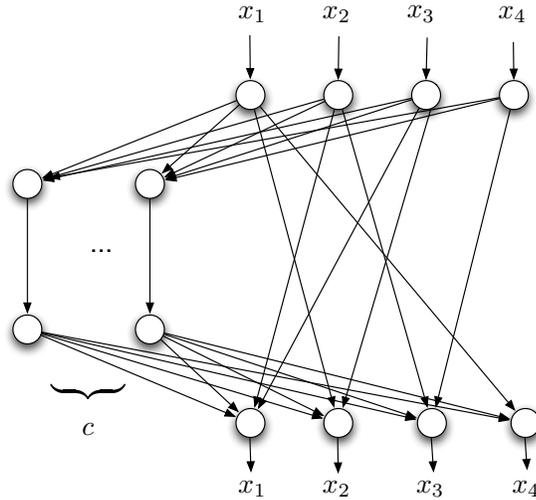


Fig. 27. An instance of the network coding problem equivalent to the instance of the index coding problem depicted in Figure 26.

Figure 27 depicts the instance of the network coding problem that is equivalent to the instance of the index coding problem presented in Figure 26, where the broadcast channel is represented by c “bottleneck” edges. The construction of this network will be detailed in Section F.

In this section we present a reduction that goes in the opposite direction, i.e., from the network coding problem to the index coding problem. This reduction shows that these two problems are equivalent for the linear case. Specifically, for each instance $\mathcal{N}(G(V, E), \delta)$ of the network coding problem, we construct a corresponding instance $\mathcal{I}_{\mathcal{N}}(Y, R)$ of the index coding problem, such that $\mathcal{I}_{\mathcal{N}}$ has an (n, q) optimal linear index code achieving the minimum number of transmissions $\lambda(n, q) = \mu(\mathcal{I}_{\mathcal{N}}) = |E|$ if and only if there exists an (n, q) linear network code for \mathcal{N} .

Definition C.1. For any instance $\mathcal{N}(G(V, E), \delta)$ of the network coding problem, define the corresponding instance $\mathcal{I}_{\mathcal{N}}(Y, R)$ of the index coding problem as follows:

1. The set of messages $Y := \{x_1, \dots, x_k\} \cup \{y_1, \dots, y_m\}$ where k and m are respectively the number of messages and the number of edges in \mathcal{N} .
2. The set of clients $R := R_1 \cup \dots \cup R_5$, where follows:
 - (a) $R_1 = \{(x_i, \{y_i\}); e_i \in S\}$
 - (b) $R_2 = \{(y_i, \{x_i\}); e_i \in S\}$
 - (c) $R_3 = \{(y_i, \{y_j; e_j \in \mathcal{P}(e_i)\}); e_i \in E \setminus S\}$
 - (d) $R_4 = \{(x(\delta(e_i)), \{y_i\}); e_i \in D\}$
 - (e) $R_5 = \{(y_i, X); i = 1, \dots, m\}$

It is easy to verify that instance $\mathcal{I}_{\mathcal{N}}(Y, R)$ satisfies $\mu(\mathcal{I}_{\mathcal{N}}) = m$.

Theorem C.1. *Let $\mathcal{N}(G(V, E), X, \delta)$ be an instance of the network coding problem, and let $\mathcal{I}_{\mathcal{N}}(Y, R)$ be the corresponding instance of the index coding problem, as defined above. Then, there exists an (n, q) optimal linear index code with normalized length m for $\mathcal{I}_{\mathcal{N}}$, if and only if, there exists a linear (n, q) network code for \mathcal{N} .*

Proof. Let $X = (x_1, \dots, x_k)$, $Y = (y_1, \dots, y_m)$ and $Z = (x_1, \dots, x_k, y_1, \dots, y_m)$. Suppose there is a linear (n, q) network code $C = \{f_e(X); f_e : (\mathbb{F}_q^n)^k \rightarrow \mathbb{F}_q^n, e \in E\}$ for \mathcal{N} over the finite field \mathbb{F}_q of size q for some integer n .

Define $g : (\mathbb{F}_q^n)^{m+k} \rightarrow (\mathbb{F}_q^n)^m$ such that $\forall Z \in (\mathbb{F}_q^n)^{m+k}, g(Z) = (g_1(Z), \dots, g_m(Z))$ where $g_i(Z) = y_i + f_{e_i}(X), i = 1, \dots, m$.

More specifically, we have

$$\begin{aligned}
g_i(Z) &= y_i + x_i & i &= 1, \dots, k, \\
g_i(Z) &= y_i + f_{e_i}(X) & i &= k + 1, \dots, m - d, \\
g_i(Z) &= y_i + x(\delta(e_i)) & i &= m - d + 1, \dots, m.
\end{aligned}$$

Next, we show that $g(Z)$ is in fact an index code for \mathcal{I}_N by proving the existence of the decoding functions. We consider the following five cases:

1. $\forall \rho = (x_i, \{y_i\}) \in R_1, \psi_\rho = g_i(Z) - y_i,$
2. $\forall \rho = (y_i, \{x_i\}) \in R_2, \psi_\rho = g_i(Z) - x_i,$
3. $\forall \rho = (y_i, \{y_{i_1}, \dots, y_{i_p}\}) \in R_3,$ since C is a linear network code for \mathcal{N} , there exists a linear function ϕ_{e_i} such that $f_{e_i}(X) = \phi_{e_i}(f_{e_{i_1}}(X), \dots, f_{e_{i_p}}(X))$. Thus, $\psi_\rho = g_i(Z) - \phi_{e_i}(g_{i_1}(Z) - y_{i_1}, \dots, g_{i_p}(Z) - y_{i_p}),$
4. $\forall \rho = (x(\delta(e_i)), \{y_i\}) \in R_4, e_i \in D, \psi_\rho = g_i(Z) - y_i,$
5. $\forall \rho = (y_i, X) \in R_5, \psi_\rho = g_i(Z) - f_{e_i}(X).$

Note that this index code is optimal since it achieves the lower bound $\mu(\mathcal{I}_N) = m = |E|$.

To prove the converse, we assume that $g(Z) = (g_1(Z), \dots, g_m(Z))$ is an optimal linear (n, q) index code for I_N over the field \mathbb{F}_q , with $g_i : (\mathbb{F}_q^n)^{m+k} \longrightarrow \mathbb{F}_q^n$. We write

$$g_i(Z) = \sum_{j=1}^k x_j A_{ij} + \sum_{j=1}^m y_j B_{ij},$$

for $i = 1, \dots, m$, and $A_{ij}, B_{ij} \in M_{\mathbb{F}_q}(n, n)$, where $M_{\mathbb{F}_q}(n, n)$ is the set of $n \times n$ matrices with elements in \mathbb{F}_q .

The functions ψ_ρ exist for all $\rho \in R_5$ if and only if the matrix $M = [B_{ij}] \in M_{\mathbb{F}_q}(nm, nm)$, which has the matrix B_{ij} as a block submatrix in the (i, j) -th position, is invertible. Define $h : (\mathbb{F}_q^n)^{m+k} \longrightarrow (\mathbb{F}_q^n)^m$, such that $h(Z) = g(ZM^{-1}), \forall Z \in (\mathbb{F}_q^n)^{m+k}$. So, we obtain

$$h_i(Z) = y_i + \sum_{j=1}^k x_j C_{ij},$$

for $i = 1, \dots, m$ and where $C_{ij} \in M_{\mathbb{F}_q}(n, n)$. We note that $h(Z)$ is a valid index code for \mathcal{I}_N . In fact, for any receiver $\rho = (x, H) \in R$ with a decoding function $\psi_\rho(g, (z)_{z \in H})$

corresponding to the index code $g(Z)$, the function $\psi'_\rho(h, (z)_{z \in H}) = \psi_\rho(hM, (z)_{z \in H})$ is a valid decoding function corresponding for the index code $h(Z)$.

For all $\rho \in R_1 \cup R_4$, ψ'_ρ exists if and only if for $i = 1, \dots, k, m-d+1, \dots, m, j = 1 \dots k$ and $j \neq i$, it holds that $C_{ij} = [0] \in M_{\mathbb{F}_q}(n, n)$ and C_{ii} is invertible, where $[0]$ denotes the all zeros matrix. This implies that

$$\begin{aligned} h_i(Z) &= y_i + x_i C_{ii}, \quad i = 1, \dots, k \\ h_i(Z) &= y_i + \sum_{j=1}^k x_j C_{ij}, \quad i = k+1, \dots, m-d \\ h_i(Z) &= y_i + x(\delta(e_i)) C_{ii}, \quad i = m-d+1, \dots, m \end{aligned} \quad (4.1)$$

Next, we define the functions $f_{e_i} : (\mathbb{F}_q^n)^k \rightarrow \mathbb{F}_q^n, e_i \in E$ as follows:

1. $f_{e_i}(X) = x_i$, for $i = 1, \dots, k$
2. $f_{e_i}(X) = \sum_{j=1}^k x_j C_{ij}$, for $i = k+1, \dots, m-d$
3. $f_{e_i}(X) = x(\delta(e_i))$, for $i = m-d+1, \dots, m$.

We will show that $C = \{f_{e_i}; e_i \in E\}$ is a linear (n, q) network code for \mathcal{N} by showing that it satisfies condition N3.

Let e_i be an edge in $E \setminus S$ with the set of parent edges $\mathcal{P}(e_i) = \{e_{i_1}, \dots, e_{i_p}\}$. We denote by $I_i = \{i_1, \dots, i_p\}$ and $\rho_i = (y_i, \{y_{i_1}, \dots, y_{i_p}\}) \in R_3$. Then, there is a linear function ψ'_{ρ_i} such that $y_i = \psi'_{\rho_i}(h_1, \dots, h_m, y_{i_1}, \dots, y_{i_p})$. Hence, there exist matrices $T_{ij}, T'_{i\alpha} \in M_{\mathbb{F}_q}(n, n)$ such that

$$y_i = \sum_{j=1}^m h_j T_{ij} + \sum_{\alpha \in I_i} y_\alpha T'_{i\alpha} \quad (4.2)$$

Substituting the expressions of the h_j 's given by Equation (4.1) in Equation (4.2), we get the following:

- T_{ii} is the identity matrix,

- $T'_{i\alpha} = -T_{i\alpha} \forall \alpha \in I_i$,
- $T_{ij} = [0] \forall j \notin I_i \cup \{i\}$.

Therefore, we obtain

$$f_{e_i} = - \sum_{\alpha \in I_i} f_{e_\alpha} T_{i\alpha}, \forall e_i \in E \setminus S,$$

and C is a feasible network code for \mathcal{N} . \square

Lemma C.2. *Let $\mathcal{N}(G(V, E), \delta)$ be an instance of the network coding problem, and let $\mathcal{I}_{\mathcal{N}}(Y, R)$ be the corresponding index problem. If there exists an (n, q) network code (not necessarily linear) for \mathcal{N} , then there exists an optimal (n, q) index code for $\mathcal{I}_{\mathcal{N}}$ with normalized length m .*

Proof. Suppose there is an (n, q) network code $C = \{f_e(X); f_e : (\Sigma^n)^k \rightarrow \Sigma^n, e \in E\}$ for \mathcal{N} over a q -ary alphabet Σ . Without loss of generality, we assume that $\Sigma = \{0, 1, \dots, q-1\}$.

Define $g : (\Sigma^n)^{m+k} \rightarrow (\Sigma^n)^m$ such that $\forall Z = (x_1, \dots, x_k, y_1, \dots, y_m) \in (\Sigma^n)^{m+k}$, $g(Z) = (g_1(Z), \dots, g_m(Z))$ with

$$g_i(Z) = y_i + f_{e_i}(X), \quad i = 1, \dots, m,$$

where “+” designates addition in $GF(q)^n$. Then, the same argument of the previous proof holds similarly here, and g is an optimal index code for $\mathcal{I}_{\mathcal{N}}$. \square

D. Connection to Matroid Theory

1. Overview of Matroid Theory

There exist many different equivalent definitions of a matroid. The following one is the most useful for our analysis.

A matroid $\mathcal{M}(Y, r)$ is a pair formed by a set Y and a function $r : 2^Y \rightarrow \mathbb{N}_0$, where 2^Y is the power set of Y and \mathbb{N}_0 is the set of non-negative integer numbers $\{0, 1, 2, \dots\}$, satisfying the following three conditions:

$$(M1) \quad r(A) \leq |A|, \forall A \subseteq Y;$$

$$(M2) \quad r(A) \leq r(B), \forall A \subseteq B \subseteq Y;$$

$$(M3) \quad r(A \cup B) + r(A \cap B) \leq r(A) + r(B), \forall A, B \subseteq Y.$$

The set Y is called the *ground set* of the matroid \mathcal{M} . The function r is called the *rank function* of the matroid. The rank $r_{\mathcal{M}}$ of the matroid \mathcal{M} is defined as $r_{\mathcal{M}} = r(Y)$.

We refer to $B \subseteq Y$ as an *independent set* if $r(B) = |B|$, otherwise, it is referred to as a *dependent set*. A maximal independent set is called a *basis*. It can be shown that all the bases in a matroid have the same cardinality. In fact, for any basis B , it holds that $r(B) = |B| = r_{\mathcal{M}}$. A minimal dependent subset $C \subseteq Y$ is referred to as a *circuit*. For each element c of C it holds that $r(C \setminus \{c\}) = |C| - 1 = r(C)$. Let $\mathfrak{B}(\mathcal{M})$ be the set of all the bases of the matroid \mathcal{M} , and $\mathfrak{C}(\mathcal{M})$ be the set of all its circuits.

Matroid theory is a well studied topic in the field of discrete mathematics. References [6] and [65] provide a comprehensive discussion of this subject. Linear and multilinear representations of matroids over finite fields are major topics in matroid theory (see [6, Chapter 6], [80], and [81]).

Definition D.1. Let $Y = \{y_1, \dots, y_m\}$ be a set whose elements are indexed by the integers from 1 to m . For any collection of m matrices $M_1, \dots, M_m \in \mathbb{M}_{\mathbb{F}}(n, k)$, and any subset $I = \{y_{i_1}, \dots, y_{i_\delta}\} \subseteq Y$, with $i_1 < \dots < i_\delta$, define

$$M_I = [M_{i_1} | \dots | M_{i_\delta}] \in \mathbb{M}_{\mathbb{F}}(n, \delta k).$$

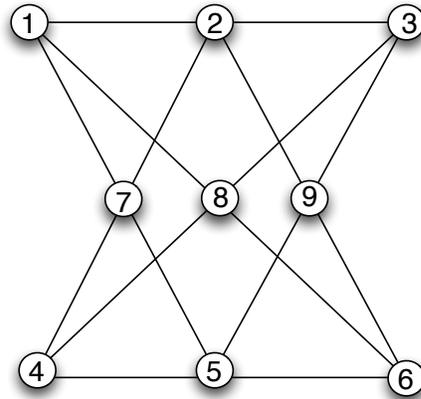


Fig. 28. A geometrical representation of the non-Pappus matroid [6, p.43]. The matroid circuits are represented by straight lines.

That is, the matrix M_I is obtained by concatenating the matrices $M_{i_1}, \dots, M_{i_\delta}$ from left to right in the increasing order of the indices i_1, \dots, i_δ .

Definition D.2. A matroid $\mathcal{M}(Y, r)$ of rank $r_{\mathcal{M}} = k$ and ground set $Y = \{y_1, \dots, y_m\}$ is said to have a *multilinear representation of dimension n* , or an *n -linear representation*, over a field \mathbb{F} , if there exist m matrices $M_1, \dots, M_m \in \mathbb{M}_{\mathbb{F}}(kn, n)$ such that

$$\text{rank}(M_I) = n \cdot r(I), \forall I \subseteq Y. \quad (4.3)$$

Linear representation corresponding to the case of $n = 1$ is the most studied case in matroid theory, see for example [6, Chapter 6]. A Multilinear representation is a generalization of this concept from vectors to vector spaces, and was discussed in [80] and [81].

Example D.1. The uniform matroid $U_{2,3}$ is defined on a ground set $Y = \{y_1, y_2, y_3\}$ of three elements, such that $\forall I \subseteq Y$ and $|I| \leq 2, r(I) = |I|$, and $r(Y) = 2$. It is easy to verify that the vectors $M_1 = [0 \ 1]^T, M_2 = [0 \ 1]^T, M_3 = [1 \ 1]^T$ form a linear

representation of $U_{2,3}$ of dimension 1 over any field. This will automatically induce a multilinear representation of dimension 2, for instance, of $U_{2,3}$ over any field:

$$M_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, M_2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, M_3 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Example D.2. The non-Pappus matroid (see e.g., [6, §1.5]) $\mathcal{M}_{np}(Y, r)$ is defined over a ground set $Y = \{y_1, \dots, y_9\}$ and can be represented geometrically as shown in Figure 28. Let $Y_0 = \{\{1, 2, 3\}, \{1, 5, 7\}, \{3, 5, 9\}, \{2, 4, 7\}, \{4, 5, 6\}, \{2, 6, 9\}, \{1, 6, 8\}, \{3, 4, 8\}\}$. The rank function of the non-Pappus matroid is

$$r(I) = \begin{cases} \min(|I|, 3) & \forall I \in 2^Y \setminus Y_0 \\ 2 & \forall I \in Y_0 \end{cases}$$

Note that Y_0 is the set of circuits of the non-Pappus matroid.

It is known from Pappus theorem [6, p.173], that the non-Pappus matroid is not linearly representable over any field. However, it was shown in [80] and [81], that it has a 2-linear representation over $GF(3)$ given below by the following 6×2 matrices M_1, \dots, M_9 :

$$[M_1 | \dots | M_9] = \begin{pmatrix} 10 & 10 & 00 & 10 & 00 & 10 & 10 & 10 & 00 \\ 01 & 01 & 00 & 01 & 00 & 01 & 01 & 01 & 00 \\ 00 & 00 & 00 & 10 & 10 & 21 & 01 & 10 & 10 \\ 00 & 00 & 00 & 02 & 01 & 20 & 12 & 02 & 01 \\ 00 & 10 & 10 & 01 & 00 & 01 & 00 & 11 & 10 \\ 00 & 01 & 01 & 21 & 00 & 21 & 00 & 10 & 01 \end{pmatrix}. \quad (4.4)$$

2. From Matroids to Index Codes

We describe now a reduction that associates to each matroid an instance of the index coding problem that captures all the dependency and independency relations of the matroid. The existence of corresponding vector linear index codes is then linked to the matroid multilinear representations.

Definition D.3. Given a matroid $\mathcal{M}(Y, r)$ of rank k and ground set $Y = \{y_1, \dots, y_m\}$, we define the corresponding index coding problem $\mathcal{I}_{\mathcal{M}}(Z, R)$ as follows:

1. $Z = Y \cup X$, where $X = \{x_1, \dots, x_k\}$,
2. $R = R_1 \cup R_2 \cup R_3$, where
 - (a) $R_1 = \{(x_i, B); B \in \mathfrak{B}(\mathcal{M}), i = 1, \dots, k\}$
 - (b) $R_2 = \{(y, C \setminus \{y\}); C \in \mathfrak{C}(\mathcal{M}), y \in C\}$
 - (c) $R_3 = \{(y_i, X); i = 1, \dots, m\}$

Note that $\mu(\mathcal{I}_{\mathcal{M}}) = m$, the cardinality of the matroid ground set.

Theorem D.3. *Let $\mathcal{M}(Y, r)$ be a matroid of ground set $Y = \{y_1, \dots, y_m\}$ and $\mathcal{I}_{\mathcal{M}}(Z, R)$ the corresponding index coding problem. The matroid \mathcal{M} has an n -linear representation over \mathbb{F}_q if and only if there exists an optimal linear (n, q) index code with normalized length m for $\mathcal{I}_{\mathcal{M}}$.*

Proof. First, we assume that all the messages in $\mathcal{I}_{\mathcal{M}}(Z, R)$ are split into n packets elements of \mathbb{F}_q , and we write $y_i = (y_{i1}, \dots, y_{in})$, $x_i = (x_{i1}, \dots, x_{in}) \in \mathbb{F}_q^n$, $\xi = (x_{11}, \dots, x_{1n}, \dots, x_{k1}, \dots, x_{kn}) \in \mathbb{F}_q^{kn}$, and $\chi = (y_{11}, \dots, y_{1n}, \dots, y_{m1}, \dots, y_{mn}, x_{11}, \dots, x_{1n}, \dots, x_{k1}, \dots, x_{kn}) \in \mathbb{F}_q^{(m+k)n}$.

Let $M_1, \dots, M_m \in \mathbb{M}_{\mathbb{F}_q}(kn, n)$ be an n -linear representation of the matroid \mathcal{M} .

Consider the following linear map $f(\chi) = (f_1(\chi), \dots, f_m(\chi))$, where

$$f_i(\chi) = y_i + \xi M_i \in \mathbb{F}_q^n, i = 1, \dots, m.$$

We claim that f is an optimal (n, q) linear index code for $\mathcal{I}_{\mathcal{M}}$. To this end, we show the existence of the decoding functions of Definition B.1 for all the clients in R :

- Fix a basis $B = \{y_{i_1}, \dots, y_{i_k}\} \in \mathfrak{B}(\mathcal{M})$, with $i_1 < i_2 < \dots < i_k$, and let $\rho_i = (x_i, B) \in R_1$, $i = 1, \dots, k$. By Equation (4.3) $\text{rank}(M_B) = kn$, hence the $kn \times kn$ matrix M_B is invertible. Thus, the corresponding decoding functions can be written as $\psi_{\rho_i} = [f_{i_1} - y_{i_1} | \dots | f_{i_k} - y_{i_k}] U_i$, where the U_i 's are the $kn \times n$ block matrices that form M_B^{-1} in the following way: $[U_i | \dots | U_k] = M_B^{-1}$.
- Let $C = \{y_{i_1}, \dots, y_{i_c}\} \in \mathfrak{C}(\mathcal{M})$, with $i_1 < i_2 < \dots < i_c$, and $\rho = (y_{i_1}, C') \in R_2$, with $C' = C - y_{i_1}$. We have $\text{rank}(M_{C'}) = \text{rank}(M_C)$ by the definition of matroid circuits. Therefore, there is a matrix $T \in \mathbb{M}_{\mathbb{F}_q}(cn - n, n)$, such that, $M_{i_1} = M_{C'} T$. Now, note that $[f_{i_2} - y_{i_2} | \dots | f_{i_c} - y_{i_c}] = \xi M_{C'}$. Therefore, the corresponding decoding function is $\psi_{\rho} = f_{i_1} - [f_{i_2} - y_{i_2} | \dots | f_{i_c} - y_{i_c}] T$.
- For all $\rho = (y_i, X) \in R_3$, $\psi_{\rho}(f, \xi) = f_i - \xi M_i$.

Now, suppose that $f(\chi) = (f_1(\chi), \dots, f_m(\chi))$, $f_i(\chi) \in \mathbb{F}_q^n$, is an optimal (n, q) linear index code for $\mathcal{I}_{\mathcal{M}}$. We will show that it induces an n -linear representation of the matroid \mathcal{M} over \mathbb{F}_q . Due to the clients in R_3 , we can assume without loss of generality that the functions $f_i(\chi)$, $i = 1, \dots, m$, have the following form

$$f_i(\chi) = y_i + \xi A_i, \tag{4.5}$$

where the A_i 's are $kn \times n$ matrices over \mathbb{F}_q . We claim that these matrices form an n -linear representation of the matroid \mathcal{M} over \mathbb{F}_q . To prove this, it suffices to show that the matrices A_i 's satisfy Equation (4.3) for all the bases and the circuits of \mathcal{M} .

Let $B \in \mathfrak{B}(\mathcal{M})$ be a basis. Then, by Equation (4.5), the clients $(x_j, B), j = 1, \dots, k$, will be able to decode their required messages iff A_B is invertible. Therefore, $\text{rank}(A_B) = nk = nr(B)$.

Let $C \in \mathfrak{C}(\mathcal{M})$ a circuit. Pick $y_{i_1} \in C$ let $C' = C - y_{i_1}$. We have $r(C') = |C| - 1 = |C'|$, i.e., C' is an independent set of the matroid, and there is a basis B of \mathcal{M} such that $C' \subseteq B$ (by the independence augmentation axiom [6, chap. 1]). Thus, from the previous discussion, $A_{C'}$ has full rank, i.e. $\text{rank}(A_{C'}) = (|C| - 1)n$. Now consider the client $\rho = (y_{i_1}, C') \in R_2$, the existence of the corresponding linear decoding function ψ_ρ implies that there exists a matrix $T \in \mathbb{M}_{\mathbb{F}}(|C|n - n, n)$ such that $A_{i_1} = A_{C'}T$. So, $\text{rank}(A_C) = \text{rank}(A_{C'}) = n(|C| - 1) = nr(C)$. \square

E. Properties of Index Codes

1. Scalar Vs. Vector Linear Codes

Index coding, as previously noted, is related to the problem of zero-error source coding with side information, discussed by Witsenhausen in [82]. Two cases were studied there, depending on whether the transmitter knows the side information available to the receiver or not. It was shown that in the former case the repeated scalar encoding is optimal, i.e., block encoding does not have any advantage over the scalar encoding. We will demonstrate in this section that this result does not always hold for the index coding problem, which can be seen as an extension of the point to point problem discussed in [82]. Alon et al. concurrently demonstrated the advantages of vector coding over scalar coding in [22] using graph theoretical techniques.

Let \mathcal{N}_1 be the M-network introduced in [4] and depicted in Figure 29. It was shown in [26] that this network does not have a scalar linear network code, but has a vector linear one of block length 2. Interestingly, such a vector linear solution does

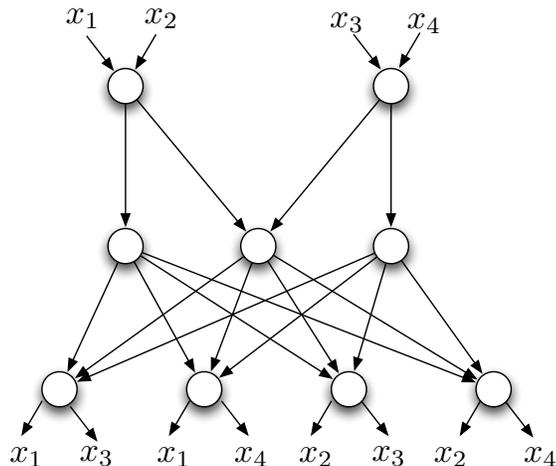


Fig. 29. The M-Network \mathcal{N}_1 introduced in [4].

not require encoding and is based on a routing scheme (see Figure 8). A more general result was proven in [26]:

Theorem E.1. *The M-network has a linear network code of block length n if and only if n is even.*

Consider the instance $I_{\mathcal{N}_1}$ of the index coding problem obtained by the construction of Definition C.1 applied to the M-network. By theorem E.1, $I_{\mathcal{N}_1}$ does not admit a scalar linear index code of normalized length $\mu(I_{\mathcal{N}_1})$. It has, however, an optimal vector linear index code of dimension 2 and normalized length $\mu(I_{\mathcal{N}_1})$ over any field. Thus, $\mathcal{I}_{\mathcal{N}_1}$ is an instance of the index coding problem where vector linear coding outperforms scalar linear one. This result can be summarized by the following corollary which follows directly from theorems C.1 and E.1.

Corollary E.2. *For $\mathcal{I}_{\mathcal{N}_1}$, $\lambda(2, 2) = \lambda^*(2, 2) < \lambda^*(1, 2)$.*

Another similar instance of the index coding problem is $\mathcal{I}_{\mathcal{M}_{np}}$, which is obtained by applying the construction of Definition D.3 to the non-Pappus matroid \mathcal{M}_{np} . Since

the non-Pappus matroid \mathcal{M}_{np} does not admit a linear representation. Theorem D.3 implies that there is also no scalar linear index code for $\mathcal{I}_{\mathcal{M}_{np}}$ that can achieve a transmission number of $\mu(\mathcal{I}_{\mathcal{M}_{np}})$. Nevertheless, the multilinear representation of the non-Pappus matroid over $GF(3)$ described in Example D.2 induces an optimal $(3, 2)$ vector linear index code for $\mathcal{I}_{\mathcal{M}_{np}}$ of normalized length $\mu(\mathcal{I}_{\mathcal{M}_{np}}) = 9$. Using Theorem D.3, we get the following corollary.

Corollary E.3. *For the instance $\mathcal{I}_{\mathcal{M}_{np}}$ of the index coding problem it holds that $\lambda^*(2, 3) = \lambda(2, 3) < \lambda^*(1, 3)$.*

2. Linear Vs. Non-Linear Codes

Linearity is a desired property for any code, including index codes. It was conjectured in [10] that scalar linear index codes over $GF(2)$ are optimal, meaning that $\lambda^*(1, 2) = \lambda(1, 2)$ for all index coding instances. Lubetzky and Stav disproved this conjecture in [12] for the *scalar linear* case by providing, for any given number of messages k and field \mathbb{F}_q , a family of instances of the index coding problem with an increasing gap between $\lambda^*(1, q)$ and $\lambda(1, q)$.

We present here another counterexample to the conjecture of [10] where even *vector linear* index codes are suboptimal. In particular, we give an instance where non-linear index codes outperform vector linear codes for any choice of field and dimension n . Our proof is based on the insufficiency of linear network codes result proved by Dougherty et al. [5]. Specifically, reference [5] showed that the network \mathcal{N}_2 depicted in Figure 30 has the following property:

Theorem E.4. *The network \mathcal{N}_2 does not admit a vector linear network code, but has a non-linear network code of block length 2 over a quaternary alphabet.*

Let $\mathcal{I}_{\mathcal{N}_2}$ be the instance of the index coding problem that corresponds to \mathcal{N}_2 ,

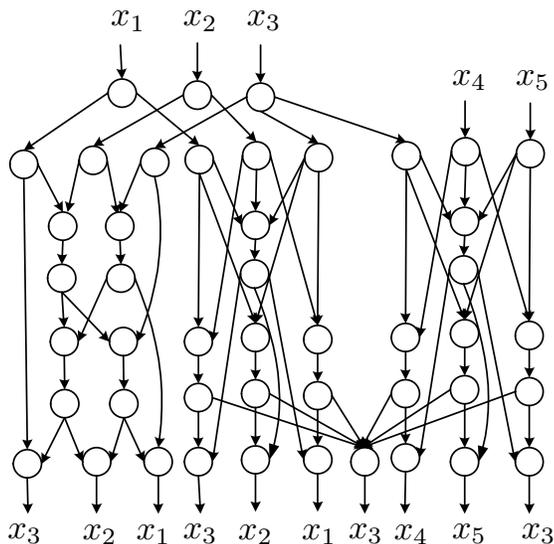


Fig. 30. The network \mathcal{N}_2 of [5]. \mathcal{N}_2 does not admit any vector linear network code, but has a non-linear one over a quaternary alphabet.

constructed according to Definition C.1. Theorem E.4 implies that $\mathcal{I}_{\mathcal{N}_2}$ does not have a linear index code of normalized length $\mu(\mathcal{I}_{\mathcal{N}_2})$. However, by Lemma C.2, a $(2, 4)$ non-linear code of \mathcal{N}_2 can be used to construct a $(2, 4)$ optimal non-linear index code for $\mathcal{I}_{\mathcal{N}_2}$ of normalized length equal to the lower bound $\mu(\mathcal{I}_{\mathcal{N}_2})$. We summarize this result by the following corollary.

Corollary E.5. *For the instance $\mathcal{I}_{\mathcal{N}_2}$ of the index coding problem, it holds that $\lambda(2, 4) = \mu(\mathcal{I}_{\mathcal{N}_2}) < \lambda^*(n, q)$, for all integers n and prime powers q .*

F. From Matroids to Networks

Dougherty et al. used in [5, 26] the results on the representability of matroids to construct the network \mathcal{N}_2 of Figure 30, which served as a counterexample to the conjecture of the sufficiency of linear network codes for achieving network capac-

ity. They also defined the concept of a matroidal network, and presented a method for constructing networks from matroids [26, Section V.B]. Given a certain matroid, they designed an instance of the network coding problem that forces the same independence relations of the matroid to exist in the set of source and edge messages. However, not all of the matroid dependency relations are reflected in this network. As a result, a linear representation for the matroid will give a linear network code for the corresponding network. However, the converse is not always true for this construction.

In this section, we present a new construction that avoids this problem and that is based on the result of Theorem D.3 which can be used as an intermediate step to build a connection between network codes and matroid linear representability. We describe below how to build a network from an index coding problem associated with a matroid, which is obtained by the construction discussed in Section 2. The reduction presented here provides a stronger connection between matroids and network codes. Specifically, for a given matroid, we construct a network such that any multilinear representation of the matroid will induce a vector linear network code for the obtained network over the same field, and vice versa. This result will permit the application of many important results on matroid linear representability to network coding theory.

Definition F.1 describes this reduction which is a generalization of the construction of the network of Figure 27. The obtained network consists of input edges representing all the messages available at the transmitter and output edges corresponding to the clients. The availability of the side information is captured by direct edges connecting a client to the corresponding nodes carrying the side information. The noiseless channel is modeled in the network by a set of “bottleneck” edges connected to all the input and output edges.

Definition F.1. Let $\mathcal{M}(Y, r)$ be a matroid of rank k defined on the set $Y = \{y_1, \dots, y_m\}$, and $\mathcal{I}_{\mathcal{M}}(Z, R)$ the corresponding Index Coding problem as described in Definition D.3. We associate to it the 6-partite network $\mathcal{N}(\mathcal{I}_{\mathcal{M}})$ over the graph $G(V, E)$ constructed as follows:

1. $V \supset V_1 \cup V_2 \cup V_3$, where $V_1 = \{s_1, \dots, s_{m+k}\}$, $V_2 = \{n'_1, \dots, n'_m\}$, and $V_3 = \{n''_1, \dots, n''_m\}$.
2. Connect each node $s_i, i = 1, \dots, k$, to an input edge carrying an information source x_i at its tail node, and each node $s_i, i = k + 1, \dots, m + k$, to an input edge carrying an information source y_i .
3. Add edges (s_i, n'_j) , for $i = 1, \dots, m + k$ and $j = 1, \dots, m$.
4. Add edges (n'_j, n''_j) for $j = 1, \dots, m$.
5. For each client $\rho = (z, H) \in R$, add a vertex n_ρ to the network, and connect it to an output edge that demands source z . In addition, for each $z' \in H$, add edge (s', n_ρ) , where $s' \in V_1$ is connected to an input edge carrying source z' .
6. For each $\rho \in R$, add edge (n''_j, n_ρ) , for $j = 1, \dots, m$.

Theorem F.1. *The matroid \mathcal{M} has an n -linear representation over \mathbb{F}_q iff the network $\mathcal{N}(\mathcal{I}_{\mathcal{M}})$ has an (n, q) vector linear network code.*

Proof. It can be easily seen that any (n, q) optimal linear index code of length m for $\mathcal{I}_{\mathcal{M}}$ will imply an (n, q) linear network code for $\mathcal{N}(\mathcal{I}_{\mathcal{M}})$, and vice versa. The proof follows, then, directly from Theorem D.3. \square

Theorem F.1 suggests that network codes can be regarded as a generalization of the concept of matroid linear representation. As a matter of fact, matroids, as dependency structures, have to satisfy constraints that do not usually apply to networks.

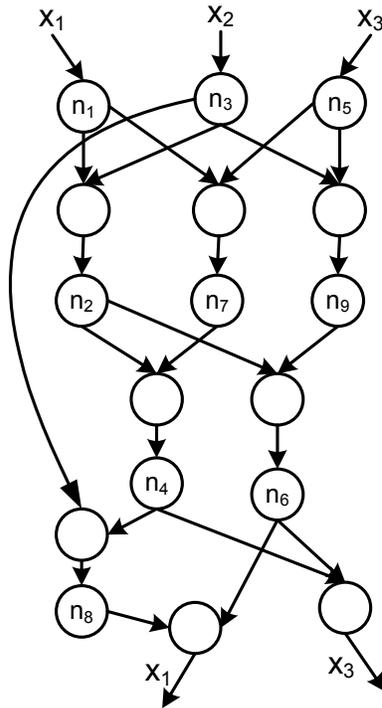


Fig. 32. A subnetwork of the network \mathcal{N}_3 .

As an application to Theorem F.1, we construct a new network that is similar to the M-network of Figure 29 in that it does not admit a scalar linear network code, but has a vector linear one. This network is depicted partially in Figure 31 and is obtained by applying the construction of Definition F.1 to the non-Pappus matroid. Node n_1 represents the clients in the set R_3 , n_2 the basis $\{1, 2, 4\}$ of the non-Pappus matroid, and n_3, n_4, n_5 the cycle $\{1, 2, 3\}$. By Theorem F.1, this network does not have a scalar linear network code, but has a vector linear code of length 2 over $GF(3)$.

The construction of networks from matroids is not unique. Next, we present another network, referred to as \mathcal{N}_3 (see Figure 32), corresponding to the non-Pappus matroid and that has the same property as the M-network.

Definition F.2. Let $S_0 = \{\{1, 2, 3\}, \{1, 5, 7\}, \{3, 5, 9\}, \{2, 4, 7\}, \{4, 5, 6\}, \{2, 6, 9\},$

$\{1, 6, 8\}, \{3, 4, 8\}$, and $S_1 = \{I \subseteq \{1, 2, \dots, 9\}; |I| = 3\} \setminus S_0$. The network \mathcal{N}_3 is obtained by adding, to the network depicted in Figure 32, a node n_I for each $I = \{i, j, k\} \in S_1$, the edges $(n_i, n_I), (n_j, n_I), (n_k, n_I)$ and three output edges outgoing from n_I , each one of them demands a different x_i .

Theorem F.2. *There exists no scalar linear network code for the network \mathcal{N}_3 over any field, but there is a $(2, 3)$ linear one.*

Proof. Let $C = \{f_e; e \in \mathcal{N}_2\}$ be a scalar linear network code for \mathcal{N}_3 over a certain field \mathbb{F} . Without loss of generality, we assume that for each node n_i of \mathcal{N}_3 , the functions associated with its output edges are identical. We define $f_i = f_e$ where e is an outgoing edge to n_i , $i = 1, \dots, 9$, and write $f_i = a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 = a_i \cdot X^T$, where $X = (x_1, x_2, x_3)$ and $a_i = (a_{i1}, a_{i2}, a_{i3})$.

Since, $\forall I = \{i, j, k\} \in S_1$, the outgoing edges to node n_I demand x_1, x_2 and x_3 , we have $\text{rank}\{a_i, a_j, a_k\} = 3$. Furthermore, from the connectivity of \mathcal{N}_3 , we deduce that a_2 should be a linear combination of a_1 and a_3 , giving $\text{rank}\{a_1, a_2, a_3\} < 3$. But $\text{rank}\{a_1, a_2, a_4\} = 3$, which implies that $\text{rank}\{a_1, a_2, a_3\} > 1$, hence $\text{rank}\{a_1, a_2, a_3\} = 2$. Similarly, $\forall \{i, j, k\} \in S_0, \text{rank}\{a_i, a_j, a_k\} = 2$.

Therefore, letting $A = \{a_1, a_2, \dots, a_9\}$, the matroid $\mathcal{M}(A, \text{rank})$ is the non-Pappus matroid shown in Figure 28 [6, p.43]. Therefore, the vectors a_i form a linear representation of \mathcal{M} over \mathbb{F} . But, by Pappus theorem [6, p.173], the non-Pappus matroid is not linearly representable over any field, which leads to a contradiction. So, \mathcal{N}_3 does not have a scalar linear solution.

Let $x_1 = (x, y), x_2 = (w, z), x_3 = (u, v) \in \mathbb{F}_3^2$. Define $f_1(X) = x_1, f_2(X) = (x + w, y + z), f_3(X) = x_2, f_4(X) = (x + u + 2z, y + 2v + w + z), f_5(X) = x_3, f_6(X) = (x + 2u + 2v + 2z, y + u + w + z), f_7(X) = (x + v, y + u + 2v), f_8(X) = (x + u + w + z, y + 2v + w), f_9(X) = (u + w, v + z)$. These functions correspond to the multilinear

(or partition) representation of the non-Pappus matroid discussed in [80, 81]. For each edge $e \in G$ outgoing from node $n_i, i = 1, \dots, 9$, define $f_e = f_i$, and for each edge $e \in D$, let $f_e = \delta(e)$. Then, $\{f_e; e \in \mathcal{N}_2\}$ is a $(2, 3)$ network code for the non-Pappus network. \square

CHAPTER V

CONCLUSION

The problem of communicating information in networks with noise-free and interference-free links was traditionally addressed using techniques inspired by the study of commodity flows in transportation networks, where the only difference taken into account between commodities and information was the possibility of duplicating information by copying. In 2000, Ahlswede *et al.* [1] presented a novel and original approach to this problem that formed the new paradigm of network coding. Network coding extends the capability of intermediate nodes in the network from mere copying to “mixing”, i.e., encoding, the different data packets received on their incoming edges. This approach was shown to produce in many scenarios a substantial throughput gain over the traditional routing and tree packing techniques [33, 34, 35]. In this Dissertation, we investigated applications of network coding for network resilience to link failures and security against wiretapping. Furthermore, we studied the relation between network coding and index coding.

First, we addressed the problem of constructing robust network codes [28, 13] achieving instantaneous recovery from single edge failures for unicast networks with non-uniform edge capacities. We demonstrated that for the case of $h = 2$ information sources at the source node, *minimal networks*, i.e., networks that do not contain redundant edges or edges with excess capacities, are characterized by a special block structure that we described. Moreover, we devised an algorithm that exploits this structure to efficiently construct robust network codes over the binary field. We have also addressed the problem of efficient resource allocation for this case. As a direction for future research, it is interesting to generalize our results to the case of multicast demands. The open question that arises in this context is whether it is possible to

characterize the structure of minimal networks for the general case of $h > 2$ or the case of multiple edge failures. Another open question is whether it is possible to construct robust network codes for $h > 2$ over a finite field whose size depends only on h and does not depend on the number of edges in the network.

Second, we considered the problem of securing a multicast network implementing network coding against a wiretapper capable of observing a limited number of edges of his choice, as defined initially by Cai and Yeung [15]. We showed that this problem can be formulated as a generalization of the wiretap channel of Type II which was introduced and studied by Ozarow and Wyner, and decomposed into two sub-problems: the first consists of designing a secure wiretap channel code, or a coset code, and the second consists of designing a network code satisfying some additional constraints. We proved that there is no penalty in adopting this separation. Moreover, this approach allowed us to derive new bounds on the required alphabet size for secure network codes. These new bounds differ from those in the literature in that they are independent from the network size and are functions of only the number of information messages and destinations. We also analyzed the performance of the proposed coset codes under various wiretapper scenarios. Many interesting questions related to this problem remain open. For instance, the bounds presented here on the code alphabet size can be large in certain cases and it is worthy to investigate whether tighter bounds exist. Another issue, which was not addressed in this dissertation, is that of designing efficient decoding algorithms at the destinations which can be very important in practical implementations. Also, the work of [56] hinted at some advantages of non-linear codes. The benefits of nonlinearity in security applications, whether at the source code or at the network code level, are still to be better understood.

Finally, we focused on the index coding problem and its relation to network

coding and matroid theory. We presented a reduction that maps an instance \mathcal{N} of the network coding problem to an instance $\mathcal{I}_{\mathcal{N}}$ of the index coding problem such that \mathcal{N} has a vector linear network code over \mathbb{F}_q if and only if there is an optimal linear index code for $\mathcal{I}_{\mathcal{N}}$ with $|E|$ transmissions \mathbb{F}_q , where E is the set of network edges. Our reduction implies that these two problems are equivalent when coding is restricted to be linear. As a consequence, many important properties of network codes carry over to index codes. In particular, using the M -network described in [4], we showed that vector linear codes outperform scalar ones. In addition, by using the results of Dougherty *et al.* in [5], we showed that non-linear codes outperform vector linear codes. We also presented a second reduction that maps an instance of the matroid representation problem to an instance of the index coding problem. In particular, for any given matroid \mathcal{M} , we constructed an instance of the index coding problem $\mathcal{I}_{\mathcal{M}}$, such that \mathcal{M} has a multilinear representation if and only if $\mathcal{I}_{\mathcal{M}}$ has a vector linear solution over the same field. Through this reduction, we were able to establish a connection, which is stronger than what is already described in the literature, between linear network codes and matroid linear representation. Furthermore, using the properties of the non-Pappus matroid, we gave more instances of the network coding and index coding problems highlighting the advantages of vector linear codes. An important question that remains open here is whether the two problems of network and index coding are also equivalent for the general non-linear case.

REFERENCES

- [1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, “Network information flow,” *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.
- [2] S. El Rouayheb, A. Sprintson, and C. Georghiades, “Simple network codes for instantaneous recovery from edge failures in unicast connections,” in *Workshop on Information Theory and its Applications*, San Diego, CA (Invited Paper), February 2006.
- [3] S. El Rouayheb, A. Sprintson, and C. N. Georghiades, “Robust network codes for unicast connections: A case study,” submitted to *IEEE/ACM Transactions on Networking*, 2009.
- [4] M. Medard, M. Effros, T. Ho, and D. R. Karger, “On coding for non-multicast networks,” in *41st Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, October 2003.
- [5] R. Dougherty, C. Freiling, and K. Zeger, “Insufficiency of linear coding in network information flow,” *IEEE Transactions on Information Theory*, vol. 51, no. 8, pp. 2745–2759, 2005.
- [6] J. G. Oxley, *Matroid Theory*, New York: Oxford University Press, January 1993.
- [7] T. Cover and A. EL-Gamal, “Capacity theorems for the relay channel,” *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 572–584, September 1979.

- [8] P. Elias, A. Feinstein, and C. E. Shannon, “A note on the maximum flow through a network,” *IEEE Transactions on Information Theory*, vol. IT-2, pp. 117–119, December 1956.
- [9] K. Jain, M. Mahdian, and M. R. Salavatipour, “Packing steiner trees,” in *14th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Baltimore, MD, January 2003.
- [10] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Ko, “Index coding with side information,” in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, Berkeley, CA, October 2006, pp. 197–206.
- [11] Y. Birk and T. Kol, “Informed-source coding-on-demand (ISCOD) over broadcast channels,” in *Proceedings of the 17th IEEE International Conference on Computer Communications (INFOCOM)*, San Francisco, CA, March 1998, vol. 3, pp. 1257–1264.
- [12] E. Lubetzky and U. Stav, “Non-linear index coding outperforming the linear optimum,” in *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, Providence, RI, October 2007, pp. 161–167.
- [13] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen, “Polynomial time algorithms for multicast network code construction,” *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973–1982, 2005.
- [14] P. A. Chou and Yunnan Wu, “Network coding for the internet and wireless networks,” *Signal Processing Magazine, IEEE*, vol. 24, no. 5, pp. 77–85, Sept. 2007.

- [15] N. Cai and R. W. Yeung, “Secure network coding,” in *IEEE International Symposium on Information Theory (ISIT)*, Lausanne, Switzerland, June 2002, p. 323.
- [16] S. El Rouayheb and E. Soljanin, “On wiretap networks II,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Nice, France, June 2007, pp. 551–555.
- [17] S. El Rouayheb, E. Soljanin, and A. Sprintson, “Secure network coding for wiretap networks of type II,” submitted to *IEEE Transactions on Information Theory*, 2009.
- [18] L. H. Ozarow and A. D. Wyner, “The wire-tap channel II,” *Bell System Technical Journal*, vol. 63, pp. 2135–2157, 1984.
- [19] L. H. Ozarow and A. D. Wyner, “Wire-tap channel II,” in *Proceedings of the EUROCRYPT 84 Workshop on Advances in Cryptology: Theory and Application of Cryptographic Techniques*. 1985, pp. 33–51, New York: Springer-Verlag.
- [20] M. Langberg, A. Sprintson, and J. Bruck, “The encoding complexity of network coding,” *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2386–2397, June 2006.
- [21] M. Langberg, A. Sprintson, and J. Bruck, “Network coding: A computational perspective,” *IEEE Transactions on Information Theory*, vol. 55, no. 1, pp. 147–157, Jan. 2009.
- [22] N. Alon, E. Lubetzky, U. Stav, A. Weinstein, and A. Hasidim., “Broadcasting with side information,” in *Proceedings of the 49th Annual IEEE Symposium on*

- Foundations of Computer Science (FOCS)*, Philadelphia, PA, October 2008, pp. 823–832.
- [23] S. El Rouayheb, A. Sprintson, and C. N. Georghiades, “On the relation between the index coding and the network coding problems,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Toronto, Canada, July 2008, pp. 1823–1827.
- [24] S. El Rouayheb, A. Sprintson, and C. N. Georghiades, “A new construction method for networks from matroids,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Seoul, Korea, June 2009, pp. 2872–2876.
- [25] S. El Rouayheb, A. Sprintson, and C. N. Georghiades, “On the index coding problem and its relation to network coding and matroid theory,” submitted to *IEEE Transactions on Information Theory*, 2009.
- [26] R. Dougherty, C. Freiling, and K. Zeger, “Networks, matroids, and non-shannon information inequalities,” *IEEE Transactions on Information Theory*, vol. 53, no. 6, pp. 1949–1969, June 2007.
- [27] 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, 2003.
- [28] R. Koetter and M. Medard, “An algebraic approach to network coding,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782 – 795, 2003.
- [29] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, “The benefits of coding over routing in a randomized setting,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Yokohama, Japan, June 2003, p. 442.

- [30] E. Erez and M. Feder, “Convolutional network codes,” in *IEEE International Symposium on Information Theory*, Chicago, IL, June 2004, p. 146.
- [31] A. I. Barbero and O. Ytrehus, “Cycle-logical treatment for ”cyclopathic” networks,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2795–2804, 2006.
- [32] T. Ho and D. S. Lun, *Network Coding: An Introduction*, Cambridge, UK: Cambridge University Press, 2008.
- [33] C. Fragouli and E. Soljanin, *Network Coding Fundamentals, Foundations and Trends in Networking*, Hanover, MA: Now Publishers Inc, 2007.
- [34] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, *Network Coding Theory*, Hanover, MA: Now Publishers Inc, June 2006.
- [35] R. Yeung, *Information Theory and Network Coding*, New York, NY: Springer, 2008.
- [36] R. Koetter and F. Kschischang, “Coding for errors and erasures in random network coding,” *IEEE Transactions on Information Theory*, vol. 54, pp. 3579–3591, August 2008.
- [37] T. Ho, M. Médard, and R. Koetter, “An information-theoretic view of network management,” *IEEE Transactions on Information Theory*, vol. 51, no. 4, pp. 1295 – 1312, April 2005.
- [38] A. E. Kamal and A. Ramamoorthy, “Overlay protection against link failures using network coding,” in *Proceedings of the 42nd Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, 2008, pp. 527–533.

- [39] S. Li and A. Ramamoorthy, “Protection against link errors and failures using network coding in overlay networks,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Seoul, Korea, July 2009, pp. 986–990.
- [40] D. S. Lun, M. Médard, R. Koetter, and M. Effros, “Further results on coding for reliable communication over packet networks,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Adelaide, Australia, September 2005, pp. 1848–1852.
- [41] D. S. Lun, M. Médard, and M. Effros, “On coding for reliable communication over packet networks,” in *Proceedings of the 42nd Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, September 2004.
- [42] C. Gkantsidis and P. R. Rodriguez, “Network coding for large scale content distribution,” in *Proceedings of the 24th IEEE International Conference on Computer Communications (INFOCOM)*, Miami, FL, March 2005, pp. 2235–2245.
- [43] D. S. Lun, M. Médard, T. Ho., and R. Koetter, “Network coding with a cost criterion,” in *The International Symposium on Information Theory and Its Applications (ISITA)*, Parma, Italy, October 2004.
- [44] S. W. Yeung and N. Cai, “On the optimality of a construction of secure network codes,” in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, Toronto, Canada, July 2008, pp. 166–170.
- [45] J. Feldman, T. Malkin, C. Stein, and R. A. Servedio, “On the capacity of secure network coding,” in *The 42nd Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, September 2004.

- [46] K. Jain, “Security based on network topology against the wiretapping attack,” *IEEE Wireless Communications*, vol. 11, pp. 68–71, February 2004.
- [47] K. Bhattad and K. R. Narayanan, “Weakly secure network coding,” in *The First Workshop on Network Coding, Theory, and Applications (NetCod)*, Riva del Garda, Italy, April 2005.
- [48] D. Silva and F. R. Kschischang, “Universal weakly secure network coding,” in *Proceedings of IEEE Information Theory Workshop*, Volos, Greece, June 2009, pp. 281–285.
- [49] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros, and D. Karger, “Byzantine modification detection in multicast networks using randomized network coding,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Chicago, IL, June 2004, p. 442.
- [50] S. Jaggi, M. Langberg, T. Ho, and M. Effros, “Correction of adversarial errors in networks,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Nice, France, June 2007, pp. 1455–1459.
- [51] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Medard, “Resilient network coding in the presence of byzantine adversaries,” in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM)*, Miami, FL, March 2005, pp. 616–624.
- [52] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, and M. Effros, “Resilient network coding in the presence of byzantine adversaries,” *IEEE Transactions on Information Theory*, vol. 54, pp. 2596–2603, June 2008.
- [53] N. Cai and R. W. Yeung, “Network error correction, part I: Basic concepts and

- upper bounds,” *Communications in Information and Systems*, vol. 6, no. 1, pp. 19–36, 2006.
- [54] N. Cai and R. W. Yeung, “Network error correction, part II: Lower bounds,” *Communications in Information and Systems*, vol. 6, no. 1, pp. 37–53, 2006.
- [55] D. Silva, R. Koetter, and F. Kschischang, “A rank-metric approach to error control in random network coding,” *IEEE Transactions on Information Theory*, vol. 54, pp. 3951–3967, Aug 2008.
- [56] D. Silva and F. R. Kschischang, “Security for wiretap networks via rank-metric codes,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Toronto, Canada, July 2008, pp. 176–180.
- [57] D. Silva and F. R. Kschischang, “Universal secure network coding via rank-metric codes,” *arXiv:0809.3546v1*, 2008.
- [58] A. Mills, B. Smith, T. C. Clancy, E. Soljanin, and S. Vishwanath, “On secure communication over wireless erasure networks,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Toronto, Canada, July 2008, pp. 161–165.
- [59] R. Peeters, “Orthogonal representations over finite fields and the chromatic number of graphs,” *Combinatorica*, vol. 16, no. 3, pp. 417–431, 1996.
- [60] Y. Wu, J. Padhye, R. Chandra, V. Padmanabhan, and P. A. Chou, “The local mixing problem,” in *The Information Theory and Applications Workshop (ITA)*, San Diego, CA, February 2006.
- [61] M. Langberg and A. Sprintson, “On the hardness of approximating the network coding capacity,” in *Proceedings of the International Symposium on Information*

- Theory (ISIT)*, Toronto, Canada, June 2008, pp. 315–319.
- [62] S. El Rouayheb, M.A.R. Chaudhry, and A. Sprintson, “On the minimum number of transmissions in single-hop wireless coding networks,” in *Proceedings of the IEEE Information Theory Workshop (ITW)*, Lake Tahoe, CA, September 2007, pp. 120–125.
- [63] M. A. R. Chaudhry and A. Sprintson, “Efficient algorithms for index coding,” in *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM)*, Phoenix, AZ, April 2008, pp. 1–4.
- [64] H. Whitney, “On the abstract properties of linear dependence,” *American Journal of Mathematics*, vol. 57, pp. 509–533, 1935.
- [65] D. J. A. Welsh, *Matroid Theory*, London: Academic Press, 1976.
- [66] Q. Sun, S. T. Ho, and S.-Y.R. Li, “On network matroids and linear network codes,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Toronto, Canada, June 2008, pp. 1833 – 1837.
- [67] W. D. Grover, *Mesh-Based Survivable Transport Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking*, New York: Prentice-Hall, 2003.
- [68] E. Ayanoglu, C.-L. I., R. D. Gitlin, and J.E. Mazo, “Diversity coding for transparent self-healing and fault-tolerant communication networks,” *IEEE Transactions on Communications*, vol. 41, no. 11, pp. 1677–1686, 1993.
- [69] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Upper Saddle River, NJ: Prentice-Hall, Inc., 1993.

- [70] G. Brightwell, G. Oriolo, and F. B. Shepherd, “Reserving resilient capacity in a network,” *SIAM Journal on Discrete Mathematics (SIDMA)*, vol. 14, no. 4, pp. 524–539, 2001.
- [71] G. Oriolo G. Brightwell and F. B. Shepherd, “Reserving resilient capacity for a single commodity with upper-bound constraints,” *Networks*, vol. 41, no. 2, pp. 87–96, 2003.
- [72] S-Y. R. Li, R. W. Yeung, and N. Cai, “Linear network coding,” *IEEE Transactions on Information Theory*, vol. 49, pp. 371–381, February 2003.
- [73] C. Fragouli and E. Soljanin, “Information flow decomposition for network coding,” *IEEE Transactions on Information Theory*, vol. 52, pp. 829–848, March 2006.
- [74] V. K. Wei, “Generalized hamming weights for linear codes,” *IEEE Transactions on Information Theory*, vol. 37, pp. 1412–1518, September 1991.
- [75] C.-K. Ngai, R. W. Yeung, and Z. Zhang, “Network generalized hamming weight,” in *Proceedings of the Workshop on Network Coding, Theory and Applications (NetCod)*, Lausanne, Switzerland, June 2009, pp. 48–53.
- [76] C.-K. Ngai and R. W. Yeung, “Secure error-correcting (SEC) network codes,” in *Proceedings of the Workshop on Network Coding, Theory and Applications (NetCod)*, Lausanne, Switzerland, June 2009.
- [77] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in the air: Practical wireless network coding,” in *Proceedings of the ACM SIGCOMM Conference on Data Communication*, New York, NY, USA, August 2006, pp. 243–254.

- [78] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, “Symbol-level network coding for wireless mesh networks,” in *ACM SIGCOMM Conference on Data Communication*, Seattle, WA, August 2008.
- [79] Y. Birk and T. Kol, “Coding-on-demand by an informed source (ISCOD) for efficient broadcast of different supplemental data to caching clients,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2825–2830, June 2006.
- [80] J. Simonis and A. Ashikhmin, “Almost affine codes,” *Designs, Codes and Cryptography*, vol. 14, pp. 179–797, 1998.
- [81] F. Matús, “Matroid representations by partitions,” *Discrete Mathematics*, vol. 203, pp. 169–194, 1999.
- [82] H. S. Witsenhausen, “The zero-error side information problem and chromatic numbers,” *IEEE Transactions on Information Theory*, vol. 22, no. 5, pp. 592–593, 1976.

APPENDIX A

PROOF OF LEMMA D.5

Let $G(V, E)$ be a simple network, and θ a flow of value three with respect to the reduced edge capacities \bar{c} . Also, let $C(V_1, V_2)$ be an (s, t) -cut $C(V_1, V_2)$ of Type 2 in $G(V, E)$. We denote by $E(C) = \{(v_1, u_1), (v_2, u_2)\}$ the set of the edges that belongs to C .

We note that nodes u_1 and u_2 must be of Type III, since only Type III nodes have incoming edges of capacity two. Thus, nodes u_1 and u_2 have two outgoing edges each of unit capacity. By Lemma C.2, one of the outgoing edges of u_1 carries a flow of one unit; we denote it by $e_1^1 = (u_1, u_1^1)$. The other outgoing edge of u_1 carries a flow of 0.5 units; we denote it by $e_1^2 = (u_1, u_1^2)$. Similarly, one of the outgoing edges of u_2 carries a flow of one unit, we denote it by $e_2^1 = (u_2, u_2^1)$. The other outgoing edge of u_2 carries a flow of 0.5 units, we denote it by $e_2^2 = (u_2, u_2^2)$. A cut of Type 2 is depicted in Figure 33.

First, suppose that none of the u_i^j nodes coincide with the terminal node t . Then, since simple networks do not have multiple edges between two nodes, we have $u_1^1 \neq u_1^2$ (i.e., u_1^1 and u_1^2 are two distinct nodes) and $u_2^1 \neq u_2^2$. Moreover, $u_1^1 \neq u_2^1$ due to flow constraints. We are then left with four possible cases.

1. All the nodes u_1^1, u_1^2, u_2^1 and u_2^2 are distinct;
2. $u_1^2 = u_2^1$ and $u_1^1 \neq u_2^2$. In other words, node u_1^2 coincides with node u_2^1 , but u_1^1 and u_2^2 are distinct;
3. $u_1^2 = u_2^1$ and $u_1^1 = u_2^2$;
4. $u_1^2 = u_2^2$.

We prove, by contradiction, that only the last two cases are possible in simple networks. Consider the first case, and suppose that all the nodes u_1^1, u_1^2, u_2^1 and u_2^2 are distinct. We choose $E_1 = \{e_1^2, e_2^2\}$. Let $G_{E_1}(\theta)$ be the residual graph of $G(V, E)$ with respect to E_1 , and G' be the subgraph of $G_{E_1}(\theta)$ induced by nodes in V_2 . Also, let E_2 be set defined by Equation (2.4). Each node in G' has out-degree at least one, for the following reasons:

1. The terminal t is of out-degree three in G' ;
2. All nodes in $G \setminus \{u_1^1, u_1^2, u_2^1, u_2^2, t\}$, have at least one incoming edge that does not belong to E_1 . In G' these edges become outgoing edges. Thus, these nodes have out-degree at least one;
3. Nodes u_1 and u_2 have respectively the following outgoing edges e_1^2 and e_2^2 that belong to E_1 .
4. Nodes u_1^1 and u_2^1 have both incoming edges in G that do not belong to E_1 . Such edges will become outgoing edges in G' .
5. Consider node u_1^2 in G . This node is either of Type II or IV. If u_1^2 is of Type IV, then u_1^2 has an incoming edge that does not belong to E_1 . In G' , this edge becomes an outgoing edge of this node. If u_1^2 is of Type II, then it has an incoming edge of capacity one and carrying a flow of 0.5 units. This incoming edge does not belong to E_1 , since u_1^2 and u_2^2 do not coincide. Thus, u_1^2 has an incoming edge that belong to E_2 , which in G' becomes an outgoing edge. The same holds for u_2^2 .

Therefore, G' includes a cycle. This cycle should include either e_1^2 or e_2^2 (or both) and, in turn, e_1^1 or e_2^1 (or both). Suppose, without loss of generality, it is e_1^1 . Thus,

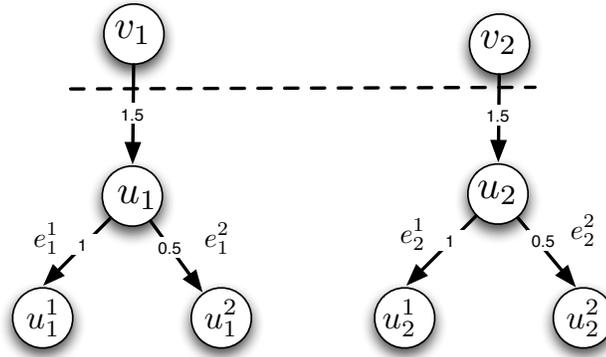


Fig. 33. A cut of Type 2. Each edge is labeled by the corresponding flow value.

the cycle should include an incoming edge of u_1^1 . In G , u_1^1 is either of Type I or IV. In both cases u_1^1 cannot have an incoming edge in E_1 since u_1^1 , u_2^2 and u_1^2 are distinct by assumption. Thus, all the incoming edges of u_1^1 belong to E_2 . Thus, G' must have a cycle that includes an edge in E_2 , i.e., a residual cycle. Hence, G' is not minimal. This completes the proof of the Lemma.

For example, consider the network depicted in Figure 34(a). The corresponding residual graph is depicted in Figure 34(a). This graph has a cycle $W = \{u_1, u_1^2, u_2^2, u_1^1, u_1\}$. This cycle includes the Type I node u_1^1 and its incoming edge (u_2^2, u_1^1) . Since $(u_2^2, u_1^1) \in E_2$, W is a residual cycle. Thus, at least two of the nodes u_i^j 's should coincide. Note that no more than two non-terminal nodes can coincide in a simple graph because of the restriction on the total degree of the nodes.

Consider now the second case, i.e., when u_1^2 and u_2^1 coincide, but u_1^1 and u_2^2 do not. We can similarly prove here, by taking $E_1 = \{e_1^2, e_2^2\}$, that the graph is not minimal. Figure 34(c) shows an example of this case. Thus, if $u_1^2 = u_2^1$, we must have $u_1^1 = u_2^2$. Also, symmetrically, if $u_1^1 = u_2^2$, we must have $u_1^2 = u_2^1$. Now, if $u_1^1 \neq u_2^2$

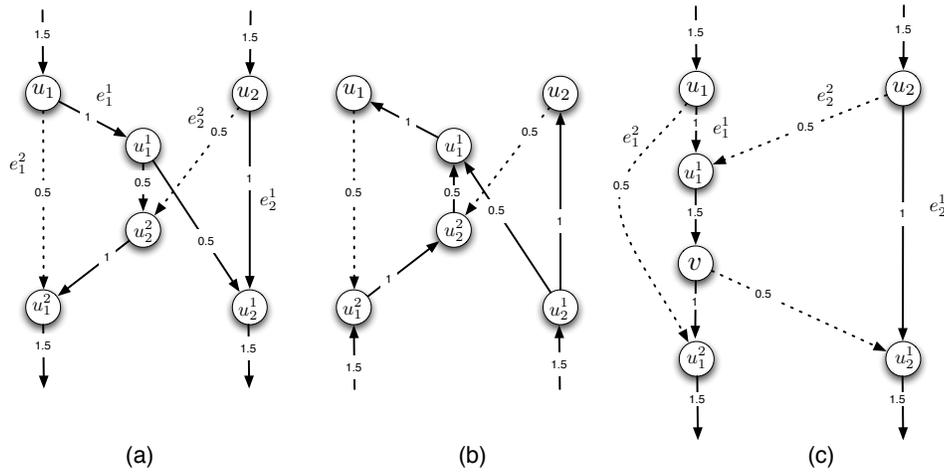


Fig. 34. Examples of subgraphs of non-minimal unicast graph that include a cut of Type 2. The labels on the edges represent the amount of flow they carry. Edge in E_1 are depicted by dashed lines. (a) An example of the case when all the nodes adjacent to u_1 and u_2 are distinct. (b) The corresponding residual graph with residual cycle $W = \{u_1, u_1^2, u_2^2, u_1^1, u_1\}$. (c) An example of the case when u_1^1 coincides with u_2^2 , but u_1^2 and u_2^1 are distinct nodes. The residual cycle in this case is $W = \{u_2, u_1^1, u_1, u_1^2, v, u_2^1, u_2\}$.

and $u_1^2 \neq u_2^1$, then by elimination, it follows that the only possible case left is when $u_1^2 = u_2^1$. In the case when u_1^1 or u_2^2 coincide with the terminal node, then we can show, by following the same steps as before, that both u_1^1 and u_2^2 coincide with the terminal node, and $u_1^2 = u_2^1$.

VITA

Salim Yaacoub El Rouayheb received his Engineering Diploma in 2002 from the Lebanese University, Faculty of Engineering, Branch II, Roumieh, Lebanon. He obtained his Master's in Computer and Communication Engineering at the American University of Beirut in 2004, and his Ph.D. degree in Electrical Engineering at Texas A&M University in 2009.

He may be reached at House of Yaacoub El Rouayheb, Koubba, Batroun, Lebanon. His email is rouayheb@gmail.com.