Random Linear Network Coding for Non-Multicast and Multi-Resolution Multicast Problems

by

Pourya Karimian

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Communications

Department of Electrical and Computer Engineering

University of Alberta

Abstract

In this dissertation, we study two network coding problems. First, we consider a class of networks that we call funnel networks. In this class of networks the total capacity of the incoming links to each intermediate node is not less than the total capacity of its outgoing links. We then prove that any feasible non-multicast problem on funnel networks is solvable by routing. This proves that a linear network coding solution exist for any non-multicast problem on funnel networks. The desirability of network coding in funnel networks may be justified by the other benefits that coding offers. However, we see that in funnel networks, the conventional random approach to linear coding fails with high probability. Hence, we provide a new random linear network coding solution for these non-multicast problems. Second, we study multicast problems in arithmetic network coding (ANC) in which, finite field arithmetic operations are replaced by real or complex arithmetic operations. A major issue in random ANC is that the condition number of the network grows quickly with the network size, hence, small errors in links can cause substantial decoding mistakes at sinks. We propose a new encoding method based on subspace coding along with a rank deficient decoding method. Our simulation results show significant improvements over conventional ANC.

"I prefer a short life with width to a narrow one with length."

- - Avicenna

Acknowledgements

I want to use this opportunity to thank everyone who has helped me in preparation of this work.

First of all, I would like to express my sincere gratitude to my supervisor, Dr. Masoud Ardakani without whom I would not be able to achieve any success during my M.Sc. program.

Second, I wish to thank the committee members, Dr. Majid Khabbazian and Dr. Chintha Tellambura, for dedicating their valuable time and energy, reading my thesis and providing comments and advice for improving it.

I also want to thank my colleagues and friends for their help and support. Friends who showed their love in both happiness and trouble and made my life very enjoyable.

Last, but not least, I want to thank my parents for their life long support and sacrifice, for which I am forever indebted to them.

Contents

List of Tables

List of Figures

List of Abbreviations

List of commonly used abbreviations

IoT	Internet of Things
RLNC	Random Linear Network Coding
ANC	Arithmetic Network Coding
MIMO.	Multiple-Input and Multiple-Output
QoS	Quality of Service
BER	Bit Error Rate

List of Symbols

List of commonly used symbols

$\mathcal{N}$	Network
v	A node in network
e	A link between two nodes
$I(v)$	Set of incoming links to node v
$O(v)$	Set of outgoing links from node v
$\mathcal{P}$	A path between two nodes
C_W	The cut-set associated with a set of nodes W
$\text{mincut}(v, v')$	Minimum value of all cut-sets between two nodes
$\mathbb{F}_{q^m}$	Finite field associated with q^m
M	Set of messages generated at the source
$D(t)$	Set of demands associated with sink t
$x(e)$	Transmitted packet over link e
$\alpha_{e',e}$	Local encoding coefficient from link e' to link e
$\beta_{i,e}$	Source encoding coefficient from ith message to link e
Y_t	Matrix of received packets at sink t
G_t	Global transfer matrix at sink t
X_s	Matrix of outgoing packets at source s
Z_t	Error matrix at sink t
T_t	Global transfer matrix from error source to sink t
$\mathcal{E}$	Matrix of noise packets on every link
r	Number of outgoing links from the source
n	Packet length
$\mathcal{P}(n, q)$	Set of all q-dimensional subspaces over $\mathbb{R}^n$
$\mathcal{C}$	A real operator channel
Λ	Input alphabet for a real operator channel
$\beta^{\perp}$	Null space associated with a subspace β
N_β	A matrix with an orthonormal basis for $\beta^{\perp}$ as its columns

Chapter 1

Motivation

Communication networks are among the most influential technologies in the past few decades. The Internet has become the crucial technology of this age. Internet of Things (IoT) is one of the major growing technologies in the last few years. The number of interconnected devices exceeded the number of people in 2011 [1] and it is estimated to reach 100 billion by 2020 [2]. This has lead to new ways of sharing information and knowledge between people or devices for various purposes such as scientific research, multimedia broadcasting, online gaming and automation.

Global network traffic is growing at an exponential rate. In a study conducted by Cisco, it is predicted that the global IP traffic will increase nearly threefold over the next five years. Table 1.1 from [3] shows the statistics of Internet users from 2000 to 2014. Handling this amount of traffic is a major issue since, available bandwidth is limited.

In addition to the bandwidth limitations, another major concern is the energy consumption of communication systems and networks, which is directly related to the amount of traffic that they handle.

Communications networks are among the major consumers of energy on the planet. According to [4], in 2013, the worldwide electricity power usage related to the ICT was around 109 GW, which represents 6% of the world electricity

Table 1.1: The statistics of global Internet users per year from 2000 to 2014

Year	Internet Users	Penetration(% of Pop)	WorldPopulation	Non-Users(Internetless)	World Pop.Change
2014	2,956,385,569	40.7 %	7,265,785,946	4,309,400,377	1.17 %
2013	2,728,428,107	38 %	7,181,715,139	4,453,287,032	1.19 %
2012	2,494,736,248	35.1 %	7,097,500,453	4,602,764,205	1.2 %
2011	2,231,957,359	31.8 %	7,013,427,052	4,781,469,693	1.21 %
2010	2,023,202,974	29.2 %	6,929,725,043	4,906,522,069	1.22 %
2009	1,766,403,814	25.8 %	6,846,479,521	5,080,075,707	1.22 %
2008	1,575,067,520	23.3 %	6,763,732,879	5,188,665,359	1.23 %
2007	1,373,226,988	20.6 %	6,681,607,320	5,308,380,332	1.23 %
2006	1,162,916,818	17.6 %	6,600,220,247	5,437,303,429	1.24 %
2005	1,030,101,289	15.8 %	6,519,635,850	5,489,534,561	1.24 %
2004	913,327,771	14.2 %	6,439,842,408	5,526,514,637	1.24 %
2003	781,435,983	12.3 %	6,360,764,684	5,579,328,701	1.25 %
2002	665,065,014	10.6 %	6,282,301,767	5,617,236,753	1.26 %
2001	502,292,245	8.1 %	6,204,310,739	5,702,018,494	1.27 %
2000	414,794,957	6.8 %	6,126,622,121	5,711,827,164	1.28 %

consumption in that year. Any solution in data transmission that reduces the traffic, data overhead, and computational complexity can have a significant impact on the global energy consumption and directly affects the environment and the future of our economy. Next generation networks are responsible to handle much larger data traffic as the number of devices and services that use these networks increase.

1.1 Network Coding Definition

In the case of communication networks, there are various applications that uses broadcasting or multicasting. Network coding has shown promising results in terms of improving the throughput and energy consumption in such cases.

In their seminal paper [5], authors describe network coding as any coding performed at network nodes, where, coding itself is an arbitrary mapping from input(s) to output(s). The basic idea is to instead of simply storing and forwarding incoming packets in intermediate network devices like routers, packets can be combined in an intelligent way such that at the destination, each user could extract the information intended to be delivered to it by some decoding function.

1.2 Benefits

Network coding has many advantages over the commonly used routing for relaying data in networks. It has been shown that network coding can improve throughput, robustness, complexity, security and more [6]. Let us consider each in turn.

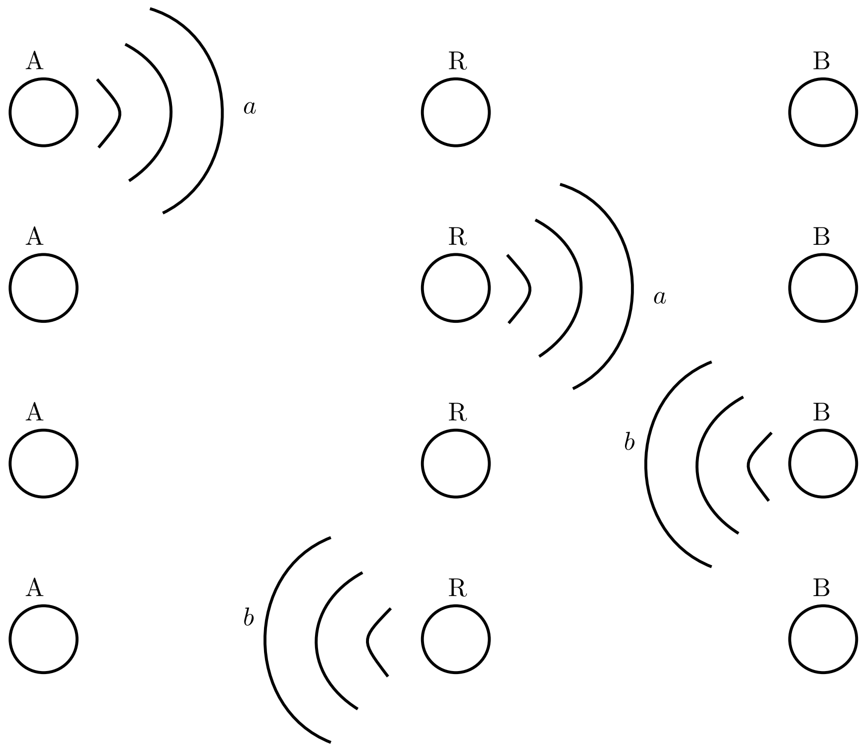
1.2.1 Throughput

By performing coding, intermediate nodes of a network can relay more information with less usage of communication links. This can be illustrated using the famous *two way relay channel*. Consider the structure depicted in Figure 1.1 in which two devices want to communicate via a relay node (e.g. a wireless access point). Using conventional methods, this can be done in four stages. However, with a simple technique, we can reduce the number of stages to three. Instead of broadcasting each message separately, the relay can get both packets in two channel use, compute and broadcast their XOR sum. Since each device has its own message, it can compute the other device's message with a simple XOR operation. Fig. 1.1 shows the stages in both scenarios.

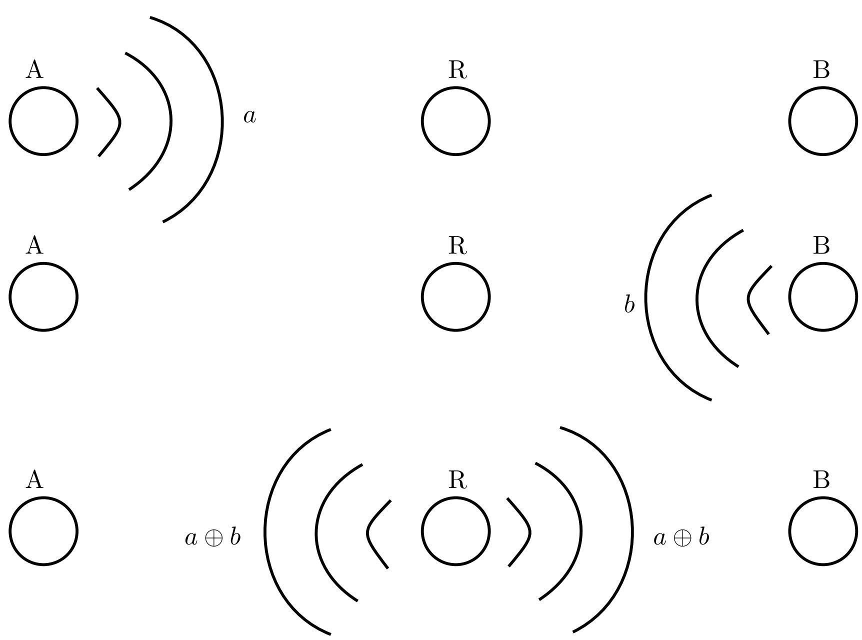
Can we do even better than three stages? By using physical layer network coding we can communicate in two stages. Although our research may be applicable to physical layer network coding, details of this method is out of the scope of this thesis. An interested reader can refer to [7] and [8] and their references for more information on this topic.

1.2.2 Robustness

Consider a reasonably large network consisting of many nodes and links. This network is responsible for relaying information from a set of sources to a set of receivers. An important issue in such networks is packet loss due to link outages and/or malicious transmission by some adversary nodes. In the case of routing, since each link is transmitting only one message at a time, the probability of loosing information due to these issues would be relatively high. In contrast, since in a network coding scenario information is encoded in several packets, we may still be able to decode information even if there are a few errors or erasures in transmission.



(a)



(b)

Figure 1.1: A simple two way relay channel: (a) Conventional relaying (b) Relaying with network coding

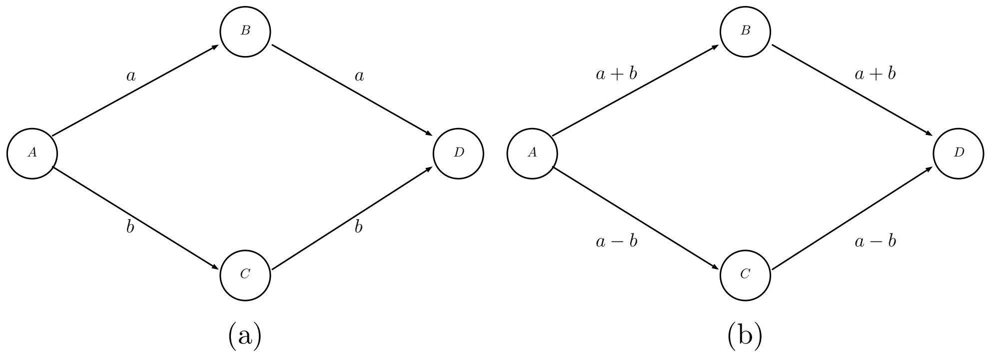


Figure 1.2: (a) Unsecured transmission. (b) Security against wiretapping.

1.2.3 Complexity

Although network coding can improve the throughput, there are cases where the optimal routing solution has the same outcome [9]. However, even in these cases, finding the optimal routing solution is an important issue that has been widely studied in the literature (see [10, 11, 12] and their references). On the other hand, as discussed in the following chapters, the optimal network coding solution can be as simple as using random linear coding at the intermediate nodes.

1.2.4 Security

While network security is an important and interesting topic of research and is often implemented regardless of using network coding or routing, it is noteworthy to mention how network coding can provide an extra layer of protection in the network. The best way to explain this is by an example. Consider the simple network of Fig. 1.2 in which node A wants to send two messages to node D. Evidently, if an adversary could access one of the links in the network, it would not gain any useful information.

1.3 Structure of the Thesis

In this thesis, we first examine a type of network in which the sum of capacities of all incoming links to an intermediate node is greater than or equal to the sum of capacities of all outgoing links from that node. Such networks are called funnel networks. We prove that for any general non-multicast problem defined on such networks, if the Min-Cut Max-Flow condition is satisfied, routing and network coding have the same throughput. However, as mentioned earlier, network coding has other benefits apart from throughput. Therefore, we propose a method based on randomized linear network coding that can be implemented for any non-multicast problem defined on funnel networks.

We then consider arithmetic network coding in which finite field operations are substituted by arithmetic operations. This kind of network coding has application in physical layer communication and multi-resolution multicast problems [13, 14]. We propose a new technique based on subspace coding [15] which will significantly reduce decoding error in arithmetic network coding.

The structure of this thesis is as follows. A detailed discussion on the graph model of networks along with basic network coding operations is presented in Chapter 2. In Chapter 3 we introduce funnel networks and present our algorithm for implementing network coding on these networks. In Chapter 4 we present our new technique based on subspace coding for arithmetic network codes along with simulation results. Finally, Chapter 5 concludes the thesis.

Chapter 2

Network Coding Preliminaries

In this chapter, we present the needed background, including modeling a communication network as a graph, finite fields, and some of the existing approaches to network coding.

2.1 Graph Model of a Network

We model a communication network by a directed acyclic graph $G(V, E)$ where, V is the set of vertices representing nodes of the network and $E \subseteq V \times V \times \mathbb{Z}_+$ is the set of directed edges.

There are two designated subsets of nodes $S, T \subseteq V$ called the set of sources and the set of sinks, respectively. A node $v \in V - (S \cup T)$ is called an intermediate node. We use $\mathcal{N}(G(V, E), S, T)$ to denote a communication network with the above properties. Although in a general settings there may be several sources in a network, the main focus of this thesis is on the networks with a single source namely, $S = \{s\}$. The task of the source is to generate information messages and inject them to the network.

A communication link between two nodes in a network can be modeled with a careful choice of directed edges in E. Each edge $e = (v, v', i) \in E$ represents a single directed communication between nodes v and v'. We call v' and v head

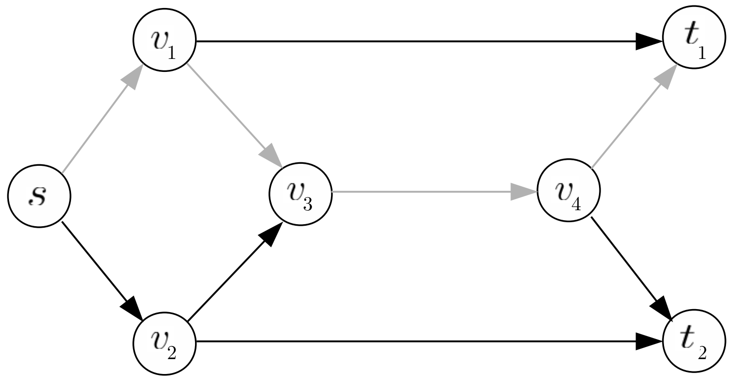


Figure 2.1: The butterfly network of Example 1

and tail of e respectively. The index i is to allow having parallel edges between two nodes. Parallel edges are sometimes used to generate a variety of channel capacities between two nodes using unit-capacity links [6]. The set of incoming edges to a node v is defined as $I_{\mathcal{N}}(v) = \{e \in E : \text{head}(e) = v\}$. Likewise, we define the set of outgoing edges from a node by $O_{\mathcal{N}}(v) = \{e \in E : \text{tail}(e) = v\}$. If no confusion arises, we drop the subscript $\mathcal{N}$ for simplicity. The *in-degree* and *out-degree* of a node v are the cardinality of its set of incoming edges ($|I(v)|$) and outgoing edges ($|O(v)|$) respectively. Without loss of generality, we assume that $I(s) = \emptyset$ and, for any sink $t \in T$, $O(t) = \emptyset$. A path $\mathcal{P}$ between two nodes v, v' is a set of edges $\{(v, v_{i_1}), (v_{i_1}, v_{i_2}), \ldots, (v_{i_j}, v')\}$ that connects v to v'.

Example 1. *Consider the network shown in Fig. 2.1. The source is depicted by the node s. Nodes v_1, v_2, v_3, v_4 are the intermediate nodes and t_1, t_2 are the sinks. A path from source s to sink t_1 is highlighted in the figure. This particular topology is known as the* ***butterfly network****.*

A *cut* between two disjoints sets of nodes $\mathcal{W}, \mathcal{W}'$ in the graph G is a partition of the vertices of the graph G into two disjoint subsets $W, V - W$ such that $\mathcal{W} \subseteq W$ and $\mathcal{W}' \subseteq V - W$. The set of edges that have one endpoint in each subset of a cut is called a *cut-set* and is denoted by C_W. The value of a cut-set is defined as the number of edges that belongs to the cut-set and is denoted

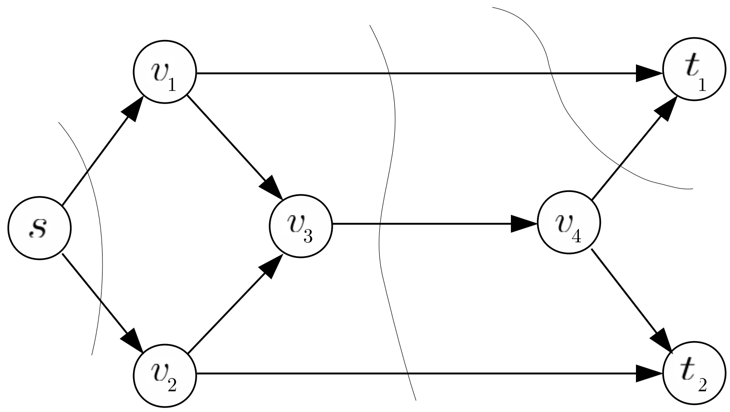


Figure 2.2: The butterfly network of Example 2 with three differnet cuts

by $\mathsf{val}(C_W) = |C_W|$. The *minimum cut* between two disjoint set of nodes in a graph is defined as the minimum value of all cut-sets between two set of nodes and is denoted by $\mathsf{mincut}(\mathcal{W}, \mathcal{W}')$.

Example 2. *Consider the butterfly network shown in Fig. 2.2. Three different cuts between source s and sink t_1 are shown in this figure. In this network, the minimum cut between s and t_1 is equal to 2.*

2.2 Finite Fields

A *finite field* or *Galois field* is a finite set of elements in which we can perform addition, subtraction, multiplication and division in a way that a certain set of rules called field axioms are satisfied. A mathematical definition of a field is given in [16] as,

Definition 1. *Consider a set F with two binary operations addition "$+$" and multiplication "$\cdot$" such that:*

1. *F is a commutative group under "$+$", with identity element 0.*

2. *The set of nonzero elements of F is also a commutative group under "$\cdot$".*

3. Multiplication is distributive over addition, i.e.,

$$\forall a, b, c \in F \quad a \cdot (b + c) = a \cdot b + a \cdot c. \tag{2.1}$$

Under these conditions F is called a field.

We call F a finite field if, the underlying set is finite. Consider a prime number q. The set of integers $\{0, \ldots, q-1\}$ together with modulo-q addition and multiplication form a finite field [16]. We denote such a field with $\mathbb{F}_q$. For every prime number q and positive integer n, we can construct finite fields of order q^n with the help of an irreducible polynomial of degree n. For these fields, we call q the characteristic of the field. Although there may be more than one finite field of order q^n (based on the choice of irreducible polynomial), all fields of that order are isomorphic. Therefore, we can use $\mathbb{F}_{q^n}$ to denote all fields of order q^n. Elements of $\mathbb{F}_{q^n}$ can be represented by polynomials of the form:

$$p(x) = p_{n-1}x^{n-1} + p_{n-2}x^{n-2} + \cdots + p_0,$$

where, $p_i \in \mathbb{F}_q$. This way, addition and subtraction in $\mathbb{F}_{q^n}$ can be performed by adding or subtracting two polynomials and reducing the coefficients of the result modulo q. As for multiplication, the result must be first reduced modulo an irreducible polynomial of degree n and then, the coefficients must be reduced modulo q.

In this thesis we are interested in another sort of multiplication in which we multiply an element of $\mathbb{F}_q$ with an element of $\mathbb{F}_{q^n}$. Let the polynomial $p(x)$ be an element of $\mathbb{F}_{q^n}$. We have,

$$\forall \lambda \in \mathbb{F}_q \quad \lambda\, p(x) = \sum_{i=0}^{n-1} (\lambda \cdot p_i) x^i, \tag{2.2}$$

where, p_i is the coefficient of x^i in $p(x)$.

Another way of representing $p(x) \in \mathbb{F}_{q^n}$ is by using row vectors of the form:

$$P = \begin{bmatrix} p_{n-1} & p_{n-2} & \dots & p_0 \end{bmatrix}.$$

This way, addition, subtraction and multiplication in (2.2) are done similar to those of vectors arithmetic with an additional reduction modulo q. We use this representation throughout the rest of this thesis.

2.3 Vector Representation of Packets

The set of messages generated at s is denoted by $M = \{m_1, m_2, \dots, m_r\}$ in which each m_i is a row vector of length n over a finite field $\mathbb{F}_q$. Associated with each sink $t \in T$ is a set of demands $D(t)$ that is a non-empty subset of M, i.e., $D(t) \subseteq M$. Each message m_i is an element of at least one $D(t)$ for some $t \in T$ otherwise, the source could exclude that message from M.

We assume that each edge e represents a communication link with unit capacity between two corresponding nodes in the sense that it can carry, free of error and with zero delay, one vector $x(e)$ from $\mathbb{F}_{q^n}$ per channel use. Such a network is called a *delay-free* network. This way, links with integer capacity can be modeled with parallel edges. Links with fractional capacity can be modeled with an arbitrary degree of accuracy by a proper choice of n and the number of channel uses.

2.4 Linear Network Coding

In a network coding scenario, each intermediate node collects the data from its incoming edges and puts a function of that data into each of its outgoing edges. In other words, for an edge $e \in O(v)$,

$$x(e) = f_e(x(e_1), x(e_2), \dots, x(e_{|I(v)|})), \tag{2.3}$$

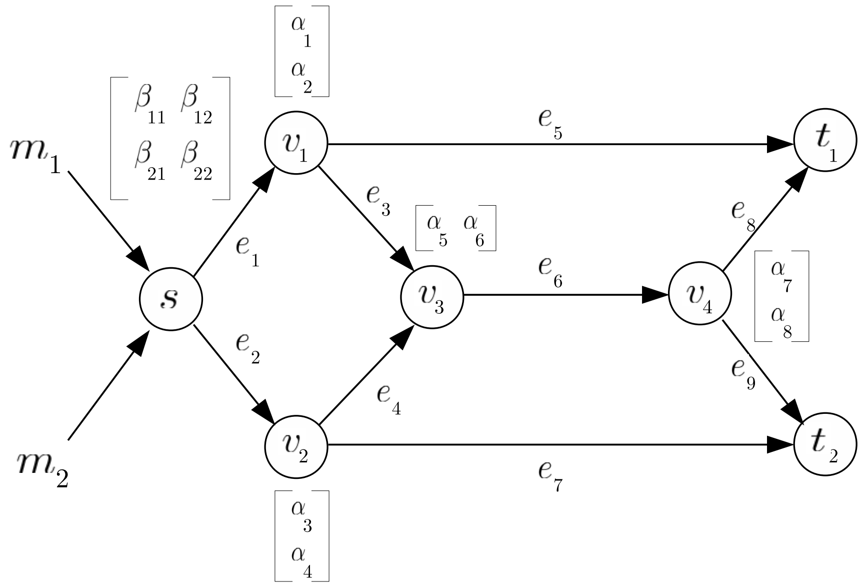


Figure 2.3: The butterfly network with local encoding coefficients.

where, e_i is an incoming edge to v and $x(e_i)$ is an element of $\mathbb{F}_{q^n}$. In the case of linear network coding, for each outgoing link, an intermediate node generates a linear combination of messages received on its incoming links, i.e., for an edge $e \in O(v)$,

$$x(e) = \sum_{e' \in I(v)} \alpha_{e',e} x(e'), \tag{2.4}$$

where, $\alpha_{e',e}$ are some elements from $\mathbb{F}_q$. In a similar fashion, the source s computes a linear combination of the messages in M for each of its outgoing links, that is, for every $e \in O(s)$,

$$x(e) = \sum_{i=1}^{|M|} \beta_{i,e} m_i. \tag{2.5}$$

Figure 2.3 shows the butterfly network with local encoding coefficients at each node. For simplicity, coefficient subscripts are changed to numbers.

A special case of linear network coding is *routing* in which each edge carries only a single message from M.

2.5 Max-flow Min-cut Condition

In a directed graph, the *max-flow min-cut theorem* states that the maximum amount of flow from a source s to a sink t is $\mathsf{mincut}(s,t)$. In a communication network, the admissible rate (information flow) between a source s and a sink t is upper bounded by $\mathsf{mincut}(s,t)$. In other words, the demands set must satisfy the max-flow min-cut bound.

The demands set satisfy max-flow min-cut bound if the union of demands sets for each subset of sinks has cardinality less than or equal to the minimum cut between the source and that subset of sinks,

$$\forall T' \subseteq T \quad \bigcup_{t\in T'} |D(t)| \leq \mathsf{mincut}(s,T'). \tag{2.6}$$

According to Menger's Theorem [17], $\mathsf{mincut}(s,t)$ is the maximum number of pairwise edge-disjoint paths from s to t.

Example 3. *Consider the butterfly network of Example 1. In this network, the minimum cut between the source and each sink is equal to 2. Moreover, the minimum cut between the source and the set of sinks is also equal to 2. This means that $D(t_1) = D(t_2) = D$ and $|D| \leq 2$. In other words, the maximum number of messages that can be transmitted from the source to any subset of sinks is bounded by 2.*

2.6 Network Equation

If we neglect the effect of delay between the source and the sinks, each sink receives a set of linear combinations of source messages from its incoming links. We show the relation between the source messages and the received vectors for sink $t \in T$ with a system of linear equations as follows,

$$Y_t = G_t \times X_s \tag{2.7}$$

In this equation, Y_t is the received vectors by sink t, G_t is the global transfer matrix between the source and the sink t, and $X_s = (P_1, P_2, \ldots, P_{|O(s)|})^\tau$ is the vector consisting of every output message $x(e),\ e \in O(s)$.

A standard state-space model for linear network coded networks is given in [18] as,

$$\begin{aligned} X &= AX + BX_s \\ X_{I(t)} &= C_t X \end{aligned} \tag{2.8}$$

where, $X = (x(e_1), x(e_2), \ldots, x(e_{|E|})^\tau$ is the network state, A is a matrix whose (i,j)th component equals to the transmittance coefficient from e_i to e_j (i.e. α_{e_i,e_j}), B is the $|E| \times |O(s)|$ matrix of coefficients from each input packet to each edge $e \in E$, and C_t is an $|I(t)| \times |E|$ matrix projecting X onto observed packets at sink t. With this state-space model we can obtain G_t, the global transfer matrix at source t, as,

$$G_t = C_t(I - A)^{-1}B. \tag{2.9}$$

It is easy to show that A is nilpotent i.e. for a sufficiently large number L, $A^L = 0$. Thus, we can obtain,

$$(I - A)^{-1} = I + A + A^2 + \cdots + A^{L-1} \tag{2.10}$$

To better illustrate this relation between the source and the sinks, we provide the following example from [18].

Example 4. *Consider the line graph depicted in Figure 2.4. This line graph is associated with the butterfly network of Figure 2.3. The source transmits two messages m_1 and m_2. By regarding the line graph as a signal flow graph with no feedback loop and m_1 and m_2 as inputs, we can find the global transfer*

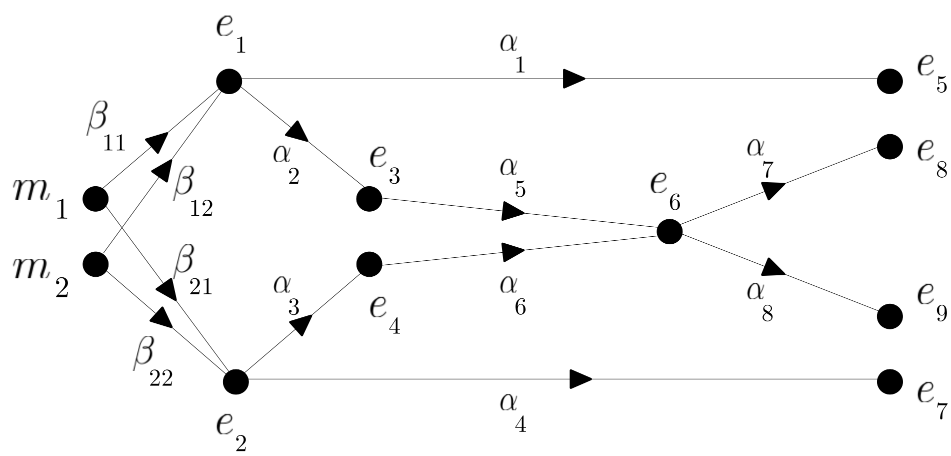


Figure 2.4: The line graph associated with the butterfly network of Fig. 2.3

matrix for each sink using the Mason's formula [19],

$$G_{t_1} = \begin{bmatrix} \beta_{11}\alpha_1 & \beta_{12}\alpha_1 \\ \beta_{11}\alpha_2\alpha_5\alpha_7 + \beta_{21}\alpha_3\alpha_6\alpha_7 & \beta_{12}\alpha_2\alpha_5\alpha_7 + \beta_{22}\alpha_3\alpha_6\alpha_7 \end{bmatrix} \tag{2.11}$$

$$G_{t_2} = \begin{bmatrix} \beta_{11}\alpha_2\alpha_5\alpha_8 + \beta_{21}\alpha_3\alpha_6\alpha_8 & \beta_{12}\alpha_2\alpha_5\alpha_8 + \beta_{22}\alpha_3\alpha_6\alpha_8 \\ \beta_{21}\alpha_4 & \beta_{22}\alpha_4 \end{bmatrix} \tag{2.12}$$

2.7 Network Problems

A network problem on $\mathcal{N}(G, S, T)$ is denoted by $(\mathcal{N}, \mathcal{D})$ where,

$$\mathcal{D} = (D(t_1), D(t_2), \ldots, D(t_{|T|})), \tag{2.13}$$

and $D(t_i)$ is the set of demands associated with sink t_i.

The network problem $(\mathcal{N}, \mathcal{D})$, is the problem of finding a network coding solution in which each sink t can receive messages in $D(t)$ with a decoding scheme. A special case of this problem is when for each $t \in T$, $D(t) = M$. Such a problem is called a multicast problem. A linear network coding solution for $(\mathcal{N}, \mathcal{D})$ exists, if there are a set of coefficients β and α such that each sink

t can decode messages in $D(t)$ from its incoming packets. A more detailed discussion on the multicast problem and its linear network coding solution is given in Section 2.8

2.8 Capacity and Code Construction

2.8.1 Multicast Capacity

For a multicast problem defined on a network $\mathcal{N}(G, S, T)$, the multicast rate $R(S, T)$ is achievable if,

$$R(S,T) < \min_{t \in T} \mathsf{mincut}(S,t). \tag{2.14}$$

This upper bound is referred to as *multicast capacity* of $\mathcal{N}$ and is achievable via network coding [5]. Moreover, Li *et al.* have shown that the multicast capacity is achievable via linear network coding, provided that the packet alphabet size is a sufficiently large finite field [20].

2.8.2 Random Linear Network Codes for Multicasting

Finding a linear network coding solution for a multicast problem has been the subject of various studies [21, 22, 23]. Notably, Ho *et al.* [24] have shown that using a sufficiently large alphabet size, if we simply choose the set of coefficients in (2.4) and (2.5) at random, sinks can decode the received packets with high probability. This method, which is called random linear network coding (RLNC), has many advantages including the ability to operate in a decentralized manner in which the intermediate nodes can construct their local encoding matrices without the need to communicate to each other, robustness to the changes of the underlying network structure (for example, the highly dynamic wireless networks). However, it also brings about an important issue that is discussed in Section 2.9

2.9 Non-coherent Transmission

A very important issue that arises by using random linear network coding is the decoding procedure. Note that in this method, neither source nor sinks have prior knowledge of the local encoding coefficients and even in some cases (e.g. dynamic networks), the underlying network structure. Two different solutions for this problem have been proposed. We describe each in turns.

2.9.1 Transmission with Headers

In this approach, the source transmits its data stream in a series of *generations*. Each generation is a cycle in which a number of packets are transmitted. During a generation, the source maps a sequence of data into $|O(s)|$ n-dimensional row vectors over $\mathbb{F}_q$ and puts them on its outgoing links. For simplicity let us assume $|O(s)| = r$. Therefore X_s in Equation 2.7 becomes,

$$X_s = \begin{bmatrix} P_1 \\ \vdots \\ P_r \end{bmatrix}, \tag{2.15}$$

where, $P_i \in \mathbb{F}_{q^n}$ is an n-dimentional vector over $\mathbb{F}_q$ that is transmitted on $e_i \in O(s)$.

With random linear network coding performed in each intermediate node, each sink t collects $|I(t)|$ randomly combined n-dimensional row vectors from its incoming links and forms them into the matrix Y_t,

$$Y_t = G_t X_s, \tag{2.16}$$

where, G_t is the $|I(t)| \times r$ global transfer matrix for t.

Assuming that t knows G_t and G_t is left invertible, it can reconstruct the injected source vectors (and subsequently the original data) by multiplying G_t^{-1}

with the received matrix $Y_{|I(t)|\times n}$ conditioned on noise matrix Z_t being zero.

A simple method for finding G_t is to set the first r by r part of X_s to the identity matrix and use the remainder of X_s as the means of communication. In other words, we set

$$P_i = (u_i, \hat{P}_i), \tag{2.17}$$

where, the *header* u_i is the ith unit vector and $\hat{P}_i$ is the *payload*. By replacing X_s in (2.7) we have,

$$\begin{aligned} Y_t &= G_t \begin{bmatrix} I_{r\times r} & X_{r\times(n-r)} \end{bmatrix} \quad (2.18) \\ &= \begin{bmatrix} G_t & G_t X \end{bmatrix}. \quad (2.19) \end{aligned}$$

Sink t can use the left part of Y_t to decode the right part if and only if G_t has rank r. In this method, by each transmission, the source s can transmit $\log_2 q^{r(n-r)}$ bits of information to sink t.

2.9.2 Transmission with Subspace

An interesting approach for transmission of data in RLNC is that instead of placing information on the elements of X_s, we can map each set of possible messages to a vector space spanned by the rows of X_s. Provided that G_t is full rank, row space of Y_t and X_t are the same. Even if G_t is not full rank, we can design a codebook of row spaces such that this kind of erasure would be potentially correctable.

Motivated by the non-coherent transmission in $\mathbb{C}$-linear multiple antenna channels [25], Koetter and Kschischang [15] proposed this method. One of the advantages of this method is the ability of designing error correction codes for network coded information without the need of an outer layer encoder/decoder. In addition, the authors showed that with this approach, the source can transmit more bits of information compared to the conventional transmission with header method.

Let W be an n-dimenstional vector space over $\mathbb{F}_q$. The row space of an $r \times n$ matrix whose elements are from $\mathbb{F}_q$ is an r-dimensional subspace of W. The set of all subspaces of W is denoted by $\mathcal{P}(W)$ The random linear network coding channel with network equation (2.7) can be regarded as an *operator channel* defined as follows.

Definition 2. *An operator channel C on vector space W is a channel whose input and output alphabet is $\mathcal{P}(W)$.*

In this approach, the source maps an input sequence of information bits to a distinct $\beta \in \mathcal{P}(W)$. Then, the source injects packets to the network in a way that β is spanned by the row space of the corresponding X_s. Since only linear combination of packets is performed at each intermediate node, regardless of the local encoding coefficients, $x(e)$ on each edge $e \in E$ is a vector in β.

With a suitable metric defined on $\mathcal{P}(W)$, each sink can decode β with a minimum distance decoding scheme. Also, the process of coding for the operator channel turns into finding a set of subspaces as a codebook that has the desired properties. Lemma 1 in [15] suggests a metric on $\mathcal{P}(W)$ as follows:

Lemma 1. *The function*

$$d(A, B) := dim(A + B) - dim(A \cap B)$$

is a metric for the space $\mathcal{P}(W)$.

Proof. The full proof is given in [15]. □

The number of distinct r-dimensional subspaces in an n-dimensional vector space ove $\mathbb{F}_q$ is given by the q-ary *Gaussian coefficient* $\begin{bmatrix} n \\ r \end{bmatrix}_q$ defined as,

$$\begin{bmatrix} n \\ r \end{bmatrix}_q := \frac{(q^n - 1)(q^{n-1} - 1)\dots(q^{n-r+1} - 1)}{(q^r - 1)(q^{r-1} - 1)\dots(q - 1)} = \prod_{i=0}^{r-1} \frac{q^{n-i} - 1}{q^{r-i} - 1}. \tag{2.20}$$

Lemma 4 in [15] gives an upper and lower bound on the Gaussian coefficient,

Lemma 2. *The Gaussian coefficient* $\begin{bmatrix} n \\ r \end{bmatrix}_q$ *satisfies*

$$q^{r(n-r)} < \begin{bmatrix} n \\ r \end{bmatrix}_q < 4q^{r(n-r)}$$

for $0 < r < n$.

Proof. For full proof, see [15] □

An interesting result of this lemma is that the number of information bits that can be transmitted with this method is more than that of the transmission with header. For more details on code construction methods and bounds, see [26, 27, 28, 29].

2.10 Arithmetic Network Coding

While in the majority of studies, linear network coding operations are performed over a finite field, there are cases where real and/or complex arithmetic operations are considered. For example, in [13], authors introduced the concept of arithmetic network coding (ANC) and have shown that it can be advantageous in some particular cases such as wireless multicast and *multi-resolution multicast*. In the latter scenario, receivers are assumed to have different download capacities. Network coding is then performed in a way that receivers with higher download capacities can receive data with a better quality of service (QoS).

2.11 Summary

In this chapter, we first presented a mathematical model based on graph theory for communication networks. Then, we provided a brief introduction to finite fields and discussed its usage to model data packets. We formally defined network coding along with different kind of network problems and discussed the theoretic bounds on achievable transmission rates in networks. We then reviewed a very useful code design for the multicast problem called RLNC. Next, we discussed about the issue of non-coherent transmission due to the usage of RLNC and presented two different solutions in the literature for this issue. Finally, we explained another network coding method called ANC in which information packets are represented by floating point real or complex numbers instead of vectors from a finite field and the network arithmetic operations are done accordingly.

Chapter 3

On Network Coding for Funnel Networks

3.1 Introduction

In this chapter, we study single-source networks in which summation of the capacities of the incoming links to each intermediate node is not less than the summation of the capacities of its outgoing links. We refer to these networks as funnel networks. This configuration has been the subject of study in graph theory as well [6, 30]. Beside the theoretical interest, this class of networks is of practical importance because such a configuration may appear as a part of larger communication networks. For example, local network between an internet service provider (ISP) and its clients or a server and clients' connection in a local area network can be modeled using this configuration.

For any general non-multicast problem on a funnel network, we prove there exists a routing solution if and only if demands satisfy the Max-flow Min-cut bound. In other words, routing and by extension, linear network coding is sufficient for this class of non-multicast problems. Note that, on a general network, non-multicast problems may not have a linear network coding solution[31, 32].

Linear network coding has many other advantages compared to routing [6].

For example, by eavesdropping on a limited number of network coded links, an adversary may not be able to recover the content of the source messages. Thus, it might be of interest to design a network code for a given problem even when routing achieves the same throughput. Unfortunately, as we show later, conventional random linear network coding performs poorly in the considered non-multicast problem. Therefore, we provide a new random design approach for funnel networks.

3.2 Related Work

Although network coding can be potentially advantageous in many network communication scenarios, there are known cases where, either network coding brings no benefit over traditional routing or there is not any straightforward method for finding a network code. Finding the limitations of network coding is of practical interest and has been the subject of many studies in the literature. For example, Li *et al.* [33, 34, 35] show that in the case of networks with undirected links, the benefit of network coding over routing in terms of throughput is at most a constant factor 2 for multicast and 1 (no additional throughput) in single unicast and broadcast.

Dougherty *et al.* [31, 32] have presented examples of network communication problems where linear network coding is insufficient to achieve the maximum transmission rate. In other words, they have shown that for a general non-multicast problem, where the receivers have different sets of demands, linear network coding is insufficient to achieve the full capacity of the network. Solving these problems often requires finding more complicated nonlinear network codes.

More recently, authors in [9] have studied benefits of network coding over routing in various scenarios. They have shown that for several networks, network coding provides no additional gain in terms of throughput or energy saving. As for code construction for non-multicast problems, a *quasi-linear network coding* method has been proposed in [36] in which, an approximate solution is found with arithmetic network coding [13]. With some careful restrictions over possible source messages and a fixed point representation, receivers can reconstruct their demanded messages. However, the achievable rates and the exact method for finding optimum codes are not discussed.

3.3 Routing vs Network Coding in Funnel Networks

In this section we prove that for funnel networks, routing achieves the Max-flow Min-cut bound of any non-multicast problem. In other words, network coding brings no additive value in terms of throughput in this class of networks.

Let us first formally define a funnel network. A funnel network is a network in which the in-degree of each node except the source is greater than or equal to its out-degree, i.e., for each $v \in V - S$, $|I(v)| \geq |O(v)|$. Theorem 1 states that any feasible non-multicast problem on funnel networks has a routing solution.

Theorem 1. *Consider an acyclic (but not necessarily connected) delay-free network $\mathcal{N}(G(V,E),S,T)$ with $S = \{s\}$ such that $\forall v \in V - S$, $|I(v)| \geq |O(v)|$. This network has a routing solution if and only if the Max-flow Min-cut bound is satisfied between the source s and each $t \in T$. Moreover, the minimum cut between s and each $t \in T$ is equal to $|I(t)|$.*

Proof. To prove this theorem we show that when the in-degree is greater than or equal to the out-degree, the Max-flow Min-cut bound simplifies to the following:

- Minimum cut between the source s and each arbitrary set of sinks, $T_i \subseteq T$, is equal to $\sum_{t \in T_i} |I(t)|$.

To do so, using induction on number of sinks $|T|$, we prove that there are $|I(t)|$ edge-disjoint paths from s to any sink $t \in T$. Moreover, these paths can be chosen in a way that any two paths ending in two different sinks are also edge-disjoint. Note that, this simultaneously proves the aforementioned simplification of Max-flow Min-cut bound and that there is a routing solution if the Max-flow Min-cut bound is satisfied. Converse statement comes from the Max-flow Min-cut theorem [37, 38]. For the base of the induction, we assume $|T| = 1$. Then we have the following lemma:

Lemma 3. *In every acyclic (but not necessarily connected) delay-free network* $\mathcal{N}(G(V, E), S, T)$ *with a single source* s *and a single sink* t *in which* $\forall v \in V - S$, $|I(v)| \geq |O(v)|$, *minimum cut between* s *and* t *is* $|I(t)|$.

Proof. We seek to prove this lemma by induction on $|I(t)|$. If $|I(t)| = 1$ we should find one path from s to t. To do this, we start by picking the only incoming edge to t, say $e_d = (v_{d-1}, t)$. Then, we pick another edge from set of incoming edges to v_{d-1} (i.e., $I(v_{d-1})$), say $e_{d-1} = (v_{d-2}, v_{d-1})$. This is possible because $|I(v_{d-1})| \geq |O(v_{d-1})|$. Then we repeat this process until we reach a point, say after d iterations, that we cannot pick any edge. This only happens if $|I(v_0)| < |O(v_0)|$ which only happens if $v_0 = s$.

For the induction step, assume that the statement of lemma is true for any network with $|I(t)| = k$. We show that this is also true for any network with $|I(t)| = k+1$. For this purpose, we choose one of the $k+1$ incoming edges of t. Similar to the above argument, we pick edges until we find a path from source to the sink. Then we remove all the edges of this path from the graph. By this removal, it is straightforward to show that the resulting graph still satisfies $|I(v)| \geq |O(v)|$ for each $v \in V - S$. Note that in this graph $|I(t)| = k$. Thus, we can find k other edge-disjoint paths in the graph. □

Lemma 3 provides the base of our induction. For induction step, we assume that for any network with $|T| = k$, minimum cut between the source s and each arbitrary set of sinks, $T_i \subseteq T$, is equal to $\sum_{t \in T_i} |I(t)|$. Consider a network $\mathcal{N}(G(V, E), \{s\}, T)$ with $T = \{t_1, t_2, \ldots, t_{k+1}\}$. We keep one of the sinks, say

t_{k+1}, and remove the rest of the sinks along with their incoming edges. The resulting network is $\mathcal{N}'(G'(V', E'), \{s\}, T')$ where, $T' = \{t_{k+1}\}$ and

$$\begin{aligned} V' &= V - (T - \{t_{k+1}\}), \\ E' &= E - \bigcup_{i=1}^{k} I(t_i). \end{aligned} \tag{3.1}$$

This network satisfies the assumptions of Lemma 3. Thus, we can find $|I(t_{k+1})|$ edge-disjoint paths from s to t_{k+1}. Define the set of these paths by P_{k+1} and all edges in these paths by E_{k+1}. Remove t_{k+1} along with all edges in E_{k+1} from $G(V, E)$ in order to get $\mathcal{N}''(G''(V'', E''), \{s\}, T'')$ such that $T'' = T - \{t_{k+1}\}$ and,

$$\begin{aligned} V'' &= V - \{t_{k+1}\}, \\ E'' &= E - E_{k+1}. \end{aligned} \tag{3.2}$$

It is straightforward to show that $G''(V'', E'')$ still satisfies $|I(v)| \geq |O(v)|$ for each $v \in V'' - \{s\}$. Note that G'' has k sinks. By induction assumption, for each $i \leq k$, there is a set of edge-disjoint paths P_i from s to t_i such that for any $i, j \leq k, i \neq j$ paths in P_i and P_j are also edge-disjoint. Note that, any pair of paths from $P_1, P_2, \ldots, P_{k+1}$ are edge-disjoint, which proves the theorem. $\square$

Note that the set of demands in Theorem 1 should not be necessarily disjoint.

3.4 Linear Network Coding for Funnel Networks

Beside increasing throughput in a communication network, (linear) network coding may offer other advantages. For instance, consider a case that an adversary has access to a limited number of links in a network and can eavesdrop

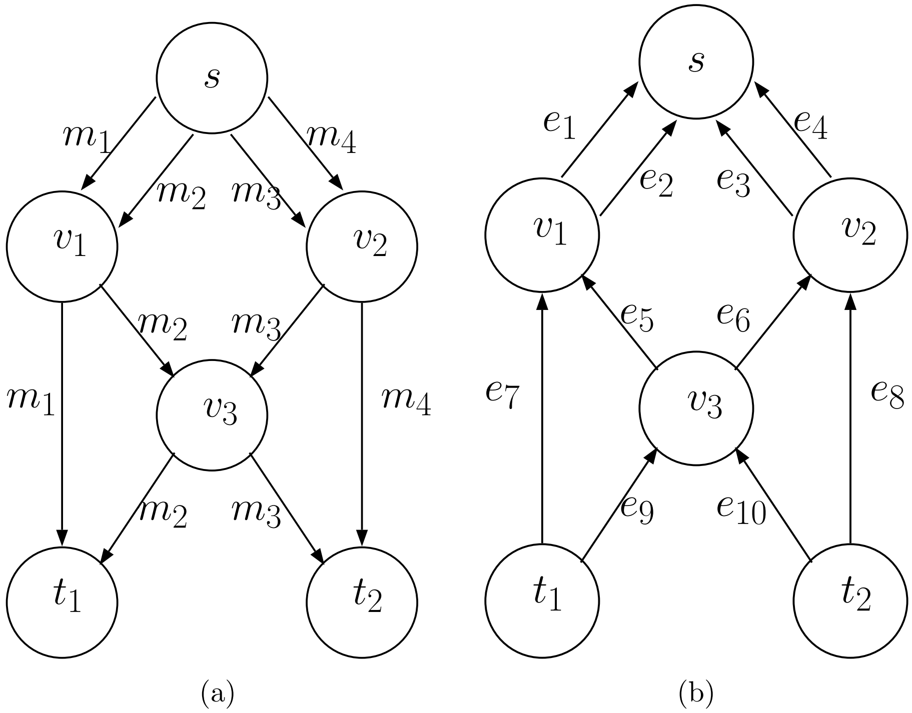


Figure 3.1: (a) A simple single-source non-multicast network with two sinks. (b) The inverse network.

the packets transmitting over those links. From a security standpoint, even without using any additional encryption, he will not be able to extract content of the original source messages unless he can collect a solvable system of equations. In such cases, a network coding solution is more preferable than routing. Another advantage of network coding over routing is adaptability to network changes. Therefore, it is suitable for dynamic structures. Here, we propose a technique based on random linear network coding to design linear network codes for networks considered in Section 3.3.

It is worth noting that the random linear approach of [24] is only suitable for multicast problems and may not work here. To better understand this point, consider the following example.

Example 5. Consider the network of Fig. 3.1a. There are four messages $M = \{m_1, m_2, m_3, m_4\}$ at the source s. Each message is a vector in $\mathbb{F}_{q^l}$. For the sinks $\{t_1, t_2\}$, we have $D(t_1) = \{m_1, m_2\}$ and $D(t_2) = \{m_3, m_4\}$. Fig.

3.1a also shows a routing solution for this particular problem. However, it is straightforward to show that if all $\alpha_{e',e}$ are chosen uniformly at random from $\mathbb{F}_q$, the probability that t_i cannot decode $D(t_i)$ is greater than $\frac{q-1}{q}$, for both $i = 1, 2$.

This means a random approach to linear network coding is not suitable here.

Instead of random selection, one can design a network coding solution which works for the given network. However, specific code designs are not flexible and cannot adjust to dynamic networks. Hence, here, we present a probabilistic method to find coefficients of a linear network code for funnel networks.

For a network $\mathcal{N}(G(V,E),S,T)$ with $|I_{\mathcal{N}}(v)| \geq |O_{\mathcal{N}}(v)|$ for all $v \in V - S$, we proceed as follows. We first define the inverse network $\mathcal{N}'(G'(V,E'),S',T')$ such that $S' = T$, $T' = S$ and $E' = \{(v',v,i) : (v,v',i) \in E\}$.

Note that this is a multi-source single-sink network which is a particular case of multi-source multicast network problems. Each sink t_i in $\mathcal{N}$ is one source in $\mathcal{N}'$. The set of messages at t_i as a source in $\mathcal{N}'$ is $M(t_i) = D(t_i)$. The set of demands at the sink s is $D(s) = M = \bigcup_{i=1}^{|S'|} M(t_i)$.

As shown in [39, Ch. 2], a random linear network code in which every $\beta'_{i,e}$ and $\alpha'_{e,e'}$ is chosen uniformly at random from $\mathbb{F}_q$ is a solution with probability arbitrary close to one if q is sufficiently large. Thus, if we let any node $v \in V - S'$ in $\mathcal{N}'$ to put a random linear combination of its incoming packets on its outgoing edges, with high probability the sink s has a solvable system of linear equations and can find all messages in $D(s)$.

By solving this system of equations, s can find $M = \{m_1, m_2, \ldots, m_r\}$. Now, coding coefficients for $\mathcal{N}$ are chosen as follows. The source s in $\mathcal{N}$ chooses the coefficients such that the output messages on $O_{\mathcal{N}}(s)$ are exactly equal to the incoming packets to the sink s in $\mathcal{N}'$. This can be done by choosing the global transfer matrix obtained from $\mathcal{N}'$ as coding coefficients for the source s in $\mathcal{N}$.

At each node $v \in V$ in $\mathcal{N}$, since $|I_{\mathcal{N}}(v)| \geq |O_{\mathcal{N}}(v)|$, if the packets on the incoming edges are exactly those output packets that were transmitted by v in $\mathcal{N}'$, we can reproduce packets on the incoming edges of v in $\mathcal{N}'$ on the outgoing links of v in $\mathcal{N}$ with high probability. This can be done by left inversion of the local transfer matrix at v in $\mathcal{N}'$[1]. Note that, if $|I_{\mathcal{N}}(v)| < |O_{\mathcal{N}}(v)|$, we may not be able to do that. Considering a topological order on the vertices V in $\mathcal{N}$, we can each time pick a vertex $v \in V$ that all of its incoming packets on $I_{\mathcal{N}}(v)$ are exactly equal to the corresponding packets in $\mathcal{N}'$ and set the coding coefficients accordingly.

This way, each sink $t \in T$ in $\mathcal{N}$ would be able to reproduce $D(t)$. To better understand this approach, we demonstrate our approach in the following example.

Example 6. The inverse of the network shown in Fig. 3.1a, is illustrated in Fig. 3.1b. The coding coefficients in the inverse network are chosen from $\mathbb{F}_3$ as follows:

$$L'_{t_1} = \begin{bmatrix} \beta'_{1,e_7} & \beta'_{2,e_7} \\ \beta'_{1,e_9} & \beta'_{2,e_9} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

$$L'_{t_2} = \begin{bmatrix} \beta'_{3,e_8} & \beta'_{4,e_8} \\ \beta'_{3,e_{10}} & \beta'_{4,e_{10}} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$$

$$L'_{v_3} = \begin{bmatrix} \alpha'_{e_9,e_5} & \alpha'_{e_{10},e_5} \\ \alpha'_{e_9,e_6} & \alpha'_{e_{10},e_6} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix}$$

$$L'_{v_2} = \begin{bmatrix} \alpha'_{e_6,e_3} & \alpha'_{e_8,e_3} \\ \alpha'_{e_6,e_4} & \alpha'_{e_8,e_4} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$L'_{v_1} = \begin{bmatrix} \alpha'_{e_5,e_1} & \alpha'_{e_7,e_1} \\ \alpha'_{e_5,e_2} & \alpha'_{e_7,e_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

[1] Note that the left inversion exists with high probability if q is chosen to be sufficiently large. Nevertheless, with careful investigation of the coding coefficients in the first step, we can make sure that the left inversion is possible.

and node s receives the following linear combinations of the messages in the inverse network

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 1 & 2 & 1 & 2 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix} \tag{3.3}$$

We can obtain the coding coefficients in the original network by using global transfer matrix given in (3.3) and inverting[2] $L_{v_1}, L_{v_2}, L_{v_3}$. Thus, we set

$$L_s = \begin{bmatrix} \beta_{1,e_1} & \beta_{2,e_1} & \beta_{3,e_1} & \beta_{4,e_1} \\ \beta_{1,e_2} & \beta_{2,e_2} & \beta_{3,e_2} & \beta_{4,e_2} \\ \beta_{1,e_3} & \beta_{2,e_3} & \beta_{3,e_3} & \beta_{4,e_3} \\ \beta_{1,e_4} & \beta_{2,e_4} & \beta_{3,e_4} & \beta_{4,e_4} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 1 & 2 & 1 & 2 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \tag{3.4}$$

$L_{v_1} = (L'_{v_1})^{-1}$, $L_{v_2} = (L'_{v_2})^{-1}$, $L_{v_3} = (L'_{v_3})^{-1}$. Thus, the global transfer matrices at sinks t_1 and t_2 are $G_{t_1} = L'_{t_1}$ and $G_{t_2} = L'_{t_2}$ respectively.

3.5 Summary

In this chapter, we showed that for any non-multicast problem on single-source networks in which summation of the capacities of the incoming links to each intermediate node is greater than or equal to the summation of the capacities of its outgoing links, there exists a routing solution that achieves the Max-flow Min-cut bound. This mean linear network coding solutions should also exist and since network coding offers other benefits, finding a linear network coding solution may be of interest for these problems. We discussed that, with high probability, conventional random linear network coding fails for the considered non-multicast problems. We then proposed a random linear network coding

[2] In general, since in-degree can be greater than the out-degree for intermediate node, the left inversion should be performed.

approach that, with high probability, achieves the Max-flow Min-cut bound.

Chapter 4

Decoding for Arithmetic Subspace Network Coding

4.1 Introduction

In this chapter, we develop a subspace *arithmetic network coding* (ANC) framework based on the subspace coding of [15, 40] for the arithmetic network coding environment of [13]. To achieve this, we first model the network input-output relation. Through studying the properties of the network noise, we develop the maximum likelihood decoding algorithm for our framework. We show that our approach can efficiently handle both the quantization noise and additive noise in physical layer. Our simulations show that our approach significantly outperforms ANC in terms of bit error rate.

By combining ANC and subspace network coding we get two main benefits. First, subspace decoding allows intermediate nodes to estimate the transmitted subspace. This way, noise is reduced, allowing for ANC to be applied to larger networks. On the other hand ANC allows for multi-resolution multicast in subspace decoding.

Two major problems in random ANC are: (i) the condition number[1] of the network grows quickly with the network size, hence, noise (e.g. quantization noise due to finite representation of real numbers) can significantly reduce the performance in larger networks; (ii) similar to other random network coding solutions, decoding cannot start unless enough number of packets are received, i.e., rank deficient decoding [42] is not possible. Since subspace network coding is an efficient solution for both error correction and rank deficiency [43], we may handle these problems by applying subspace coding to ANC. However, existing subspace network coding solutions are based on finite field operations. That is, they cannot be used with ANC. Some of the difficulties of applying subspace coding to ANC are: (i) with real arithmetic there are infinite subspaces to chose from; (ii) the effect of noise is on all links, where the noise strength gradually increases every time packets are coded at intermediate nodes; and (iii) the existing decoding algorithms for subspace decoding are not directly applicable to arithmetic operations.

4.2 Related Work

The idea of using real fields instead of finite fields for linear network coding was first introduced in [14] and further developed in [13] and it has been the subject of many studies since then. In [44], authors presented a compression scheme called quantized network coding for sparse sources by combining real network coding and concepts of compressed sensing.

Due to the similarities between the end-to-end channel model of ANC and non-coherent MIMO systems, similar studies have been done in the field of non-coherent MIMO channels (for example, [25, 45, 46, 47, 48]). However, the nature of our communication channel is very different. Hence, we cannot

[1]Condition number of a system of linear equations is a measure to describe how sensitive the solution is to changes in the inputs[41].

directly apply their results to our problem. Also, in communication networks, since we have control over some properties of the network (e.g. local encoding vectors), we have more freedom in designing encoding/decoding procedures which in turn can be used in our advantage.

4.3 System Model

In this section we present a system model slightly different from the model described in Chapter 2. The graph model is still applicable in this context. However, in certain scenarios including ANC, we must consider some kind of error in transmission. For example, in an ANC scenario, each link carries a vector of floating point real numbers instead of elements in $\mathbb{F}_q$. Intermediate nodes collect these vectors from their incoming links and perform finite precision linear arithmetic operations on them. We can regard this finite precision operations as some kind of quantization before transmission over a communication link. This rounding operation creates a quantization error in each link that propagates and accumulates through the network. Thus we must improve our model by considering this quantization error (and any other types of error for that matter). Addressing errors in a network coding scenario is of utmost importance because, even a single erroneous packet may propagate through the network and cause more errors in other messages due to message combining operations of network coding.

4.3.1 Error Model

We model error by adding an imaginary source s' to the network. For each link $e \in E$, s' is connected to $tail(e)$. This way, we can rewrite 2.7 as,

$$Y_t = G_t \times X_s + Z_t \tag{4.1}$$

where, Z_t is the matrix of noise packets received at sink t. Based on the above noise model Z_t can be written as,

$$Z_t = T_t \times \mathcal{E}, \tag{4.2}$$

where, $\mathcal{E}$ is the matrix modeling error (noise) injected by the imaginary source s' to each $e \in E$ and T_t is the transfer matrix between s' and t.

As mentioned earlier, we consider single source networks in this thesis, i.e., $S = \{s\}$. Also, without the loss of generality, we assume that the network has only one sink, namely t (the results provided in this chapter can be easily generalized to networks with multiple sinks). Hence, we can rewrite (2.7) as,

$$Y_{m\times n} = G_{m\times r} X_{r\times n} + Z_{m\times n}, \tag{4.3}$$

$$Z = T_{m\times |E|}\mathcal{E}_{|E|\times n}, \tag{4.4}$$

where, m is the number of incoming edges to sink t, r is the number of outgoing edges from source s, n is the length of network packets, G is the global transfer matrix from source s to the sink t, T is the transfer matrix from the imaginary error source s' to t, X is the input matrix generated by the source s, Y is the received matrix by t and $\mathcal{E}$ is the noise matrix, modeling the overall effect of different kinds of noise. Be it quantization noise or physical layer additive noise, elements of $\mathcal{E}$ are i.i.d zero mean random variables.

Throughout this chapter, we assume that local encoding vectors at each intermediate node are chosen randomly. Hence, the channel is non-coherent, i.e., neither source nor the sink has prior information about G and T. Therefore, their elements are random variables.

Distribution of the elements of G and T depends on the design procedure of local encoding vectors at each intermediate node and characteristics of the network. Assuming that the network is dynamic and sufficiently large, we can reasonably assume elements of T are i.i.d zero mean random variables.

In Section 4.4, we show that the elements of Z are uncorrelated. In addition, since each element of Z is a linear combination of elements of $\mathcal{E}$, if the network is sufficiently large, elements of Z can be approximated by i.i.d zero mean Gaussian random variables.

4.3.2 Real Operator Channel

Let $\mathcal{P}(n, q)$ denote the set of all q-dimensional subspaces over the vector space $\mathbb{R}^n$. Similar to the finite field operator channel discussed in Chapter 2, we define a *real operator channel* for the purpose of arithmetic network coding as follows:

Definition 3. *A real operator channel $\mathcal{C}$ is a channel with a finite set $\Lambda \subset \mathcal{P}(n, q)$ as input alphabet and $\mathcal{P}(n, q')$ as output alphabet where, $q' \leq q$*

In a sense, a real operator channel acts like a *discrete-input continuous-output* channel. Every element $\beta \in \mathcal{P}(n, q)$ can be described using a q by n matrix B which we call the *base matrix* where its rows are a basis of β. Associated with each subspace β is another subspace $\beta^{\perp}$ that contains all vectors in $\mathbb{R}^n$ that are orthogonal to every vector in β. In other words, $\beta^{\perp}$ is the null space of β. We define N_{β} as an n by $n - q$ matrix where its columns are an orthonormal basis for $\beta^{\perp}$.

4.4 Proposed Method

As discussed earlier in Chapter 2, a method for transmitting information in a non-coherent random linear network coding scenario is transmission with headers. In this method, the first $r \times r$ part of the input matrix in (4.3) is set to be the identity matrix and information is then put on the remaining $(n - r) \cdot r$ elements of the matrix. After transmission, each sink can retrieve the global encoding matrix from the output and solve a system of linear equations (perform a matrix inversion) in order to decode the original messages. However, as we demonstrate through simulation in Section 4.5, condition number of this system of linear equations grows with the size of the network [13] which means small noises can significantly deteriorate the performance. We propose two alternative coding method in this section.

4.4.1 Scale and Forward Subspace Coding

If we consider the row space of X and Y in (4.3) as the means of communication between the source and the sink in the network, we can model (4.3) as a real operator channel $\mathcal{C}$ with input alphabet $\Lambda \subset \mathcal{P}(n, q)$ and output alphabet $\mathcal{P}(n, k)$ where, q is equal to the rank of network input X and k is equal to the rank of the global encoding matrix G. Note that if k is less than q, erasure has happened and if k is equal to q and m is greater than q, we can consider additional rows as diversity and use them to reduce the effect of noise. Without loss of generality, unless otherwise mentioned, we assume q is equal to the number of outgoing edges from the source s i.e. $q = r$.

A codebook C for $\mathcal{C}$ is defined as a finite set of base matrices. There is a one to one correspondence between elements of C and Λ where, each element $B \in C$ is a base matrix for a distinct input symbol $\beta \in \Lambda$. Although the process of codebook design is an interesting and important topic of research, it is out of the scope of this thesis. Instead, we put our main focus on the decoding. Therefore, we assume n, r are predefined and C is given.

In a single channel use, the source encodes a sequence of information bits into a distinct input symbol $\beta \in \Lambda$ and sets its corresponding base matrix B as the network input X_s. Each intermediate node $v \in V - (S \cup T)$ simply forwards a combination of messages from its incoming edges to each outgoing edge according to

$$x(e') = \lambda_{e'} \sum_{e \in I(v)} \alpha_{e',e} x(e), \qquad (4.5)$$

where $\alpha_{e',e}$s are zero mean uniformly random coefficients and $\lambda_{e'}$ is a scaling factor to insure the resulting numbers are in the range of quantization. Upon receiving Y, the sink estimates the transmitted input symbol according to the following decision rule

$$\tilde{\beta} = \underset{\beta \in \Lambda}{\operatorname{argmax}} \; P(\beta|Y) \tag{4.6}$$

$$= \underset{\beta \in \Lambda}{\operatorname{argmax}} \; \frac{f(Y|\beta)P(\beta)}{f(Y)}, \tag{4.7}$$

where $f(Y)$ denotes the joint probability density function of the elements of Y. If we assume that all elements of Λ are chosen with equal probability, we have

$$\tilde{\beta} = \underset{\beta \in \Lambda}{\operatorname{argmax}} \; f(Y|\beta). \tag{4.8}$$

On the other hand, by conditioning on β, noise can be regarded as a combination of parallel and perpendicular components to the transmitted subspace β. Since the parallel component of the noise does not change the subspace of the transmitted codeword, we may just consider the perpendicular component in decoding. Using (4.3) and (4.4), we can define $Z_{\perp\beta}$ as follows

$$Z_{\perp\beta} := YN_\beta \tag{4.9}$$

$$= GBN_\beta + ZN_\beta. \tag{4.10}$$

Now we can rewrite (4.8) as follows

$$\tilde{\beta} = \underset{\beta \in \Lambda}{\operatorname{argmax}} \; f(Z_{\perp\beta}). \tag{4.11}$$

Calculating the above expression depends on the distribution of Z. The following lemma states that elements of Z are uncorrelated.

Lemma 4. *Let Z denote the product of two matrices $T_{m\times n}$ and $E_{n\times n'}$ with independent zero mean random elements. Then, the elements of Z are uncor-*

related. In addition, if elements of T and E are i.i.d with variances σ_T^2 and σ_E^2 respectively, then, elements of Z have equal variance defined as

$$\sigma_Z^2 = \sum_{k=1}^{n} \sigma_T^2 \sigma_E^2.$$

Proof. Let $T = [t_{ij}]_{m\times n}$ and $E = [e_{ij}]_{n\times n'}$. We prove that $\mathrm{Cov}(z_{ij}, z_{i'j'}) = 0$ if $(i,j) \neq (i',j')$.

$$\begin{aligned} E[z_{ij}] &= E\left[\sum_{k=1}^{n} t_{ik} e_{kj}\right] \\ &= \sum_{k=1}^{n} E[t_{ik}]E[e_{kj}] = 0. \end{aligned}$$

Therefore

$$\begin{aligned} \mathrm{Cov}(z_{ij}, z_{i'j'}) &= E[(z_{ij} - \bar{z}_{ij})(z_{i'j'} - \bar{z}_{i'j'})] \\ &= E[z_{ij} z_{i'j'}] \\ &= E\left[\left(\sum_{k=1}^{n} t_{ik} e_{kj}\right)\left(\sum_{k=1}^{n} t_{i'k} e_{kj'}\right)\right] \\ &= \sum_{k=1}^{n}\sum_{k'=1}^{n} E[t_{ik} t_{i'k'}]E[e_{kj} e_{k'j'}]. \end{aligned}$$

It is evident that the above expression has a nonzero value only if both $i = i'$ and $j = j'$. Also, if elements of T and E are i.i.d with variances σ_T^2 and σ_E^2 respectively, then

$$\begin{aligned} \sigma_Z^2 &= E[(z_{ij} - \bar{z}_{ij})^2] \\ &= \sum_{k=1}^{n}\sum_{k'=1}^{n} E[t_{ik} t_{ik'}]E[e_{kj} e_{k'j}] \\ &= \sum_{k=1}^{n} E[t_{ik}^2]E[e_{kj}^2] = \sum_{k=1}^{n} \sigma_T^2 \sigma_E^2. \end{aligned}$$

□

In addition, by assuming that the network is sufficiently large, since each element of Z is a linear combination of noise in all of the links in the network, and because elements of $\mathcal{E}$ satisfy Lyapunov condition, we expect each element of Z to have a Gaussian distribution. Therefore, according to the above lemma, we can reasonably assume that the elements of Z are i.i.d zero mean Gaussian random variables. Thus, projecting Z into a subspace of $\mathbb{R}^n$ does not change the distribution of the elements of Z. Therefore

$$\tilde{\beta} = \underset{\beta\in\Lambda}{\operatorname{argmax}} \prod_{i=1}^{m}\prod_{j=1}^{n-k} f(Z_{\perp\beta}(i,j)) \tag{4.12}$$

$$= \underset{\beta\in\Lambda}{\operatorname{argmin}} \sum_{i=1}^{m}\sum_{j=1}^{n-k} Z_{\perp\beta}(i,j)^2 \tag{4.13}$$

$$= \underset{\beta\in\Lambda}{\operatorname{argmin}} \|Z_{\perp\beta}\|_F, \tag{4.14}$$

where $Z_{\perp\beta}(i,j)$ denotes the element in the ith row and jth column of $Z_{\perp\beta}$ and $\|\cdot\|_F$ is the Frobenius norm of a matrix.

4.4.2 Project and Forward Subspace Coding

In order to reduce the effect of noise, we can let the intermediate nodes who have enough received vectors to first estimate which subspace is transmitted by the source. These nodes can then project their received vectors onto this subspace to remove the noise of previous links. Hence, we propose the following procedure for relaying information at the intermediate nodes. First, based on the number of incoming edges, we partition the set of intermediate nodes into two subsets V_1 and V_2 defined as

$$V_1 = \{v \in V - (S \cup T)|\ |I(v)| \geq \tau\} \tag{4.15}$$

and

$$V_2 = V - V_1 - (S \cup T).$$

The nodes in V_2 cannot decode the transmitted subspace since they do not receive enough incoming packets. Hence, each node $v \in V_2$ performs only scale and forward according to (4.5). Nodes in V_1 on the other hand, can estimate the network input β according to (4.14). These nodes then can construct their output vectors as follows

$$x(e') = \lambda_{e'} \sum_{e \in I(v)} \alpha_{e',e} \, \mathtt{proj}_{\tilde{\beta}} \left(x(e) \right), \tag{4.16}$$

where $\lambda_{e'}$ is the scaling factor and $\mathtt{proj}_{\tilde{\beta}}(\cdot)$ denotes the projection of a vector in the estimated subspace $\tilde{\beta}$.

4.5 Simulation Results

4.5.1 Example

In order to examine the effectiveness of our method, we present a simple codebook design method for the line and stacked butterfly networks [13] depicted in Fig. 4.1. In the stacked butterfly network, $q = k = 2$. For the case of line network, $m = k$ and we set $q = 2$. We use additional rows to examine the effect of diversity in our proposed decoding method.

We design our codebook for the simple case of $n = 2$. Thus, the input alphabet Λ is a set of 2-dimensional planes in the 3-dimensional space. The

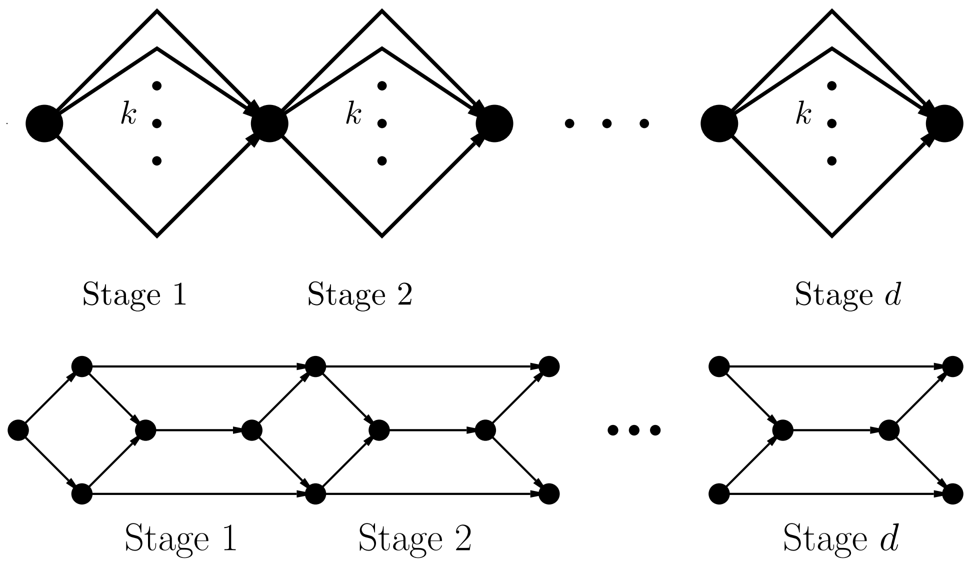


Figure 4.1: (a) Line network with d stages. (b) Stacked butterfly network with d stages.

codebook is a set of 2^l base matrices constructed according to

$$C = \{V_t \mid t = 1, \ldots 2^l, \},$$
$$V_t = \begin{bmatrix} 0 & 0 & 1 \\ -sin(\frac{t*\pi}{2^l}) & cos(\frac{t*\pi}{2^l}) & 0 \end{bmatrix} \tag{4.17}$$

Note that each element of the codebook is a rotation of the yz-plane around the z-axis. This procedure is similar to the PSK modulation. Encoding is done in a way that the l-bits sequences assigned to adjacent symbols are different in only one bit. In the next section, for these settings, we simulate the bit error rate.

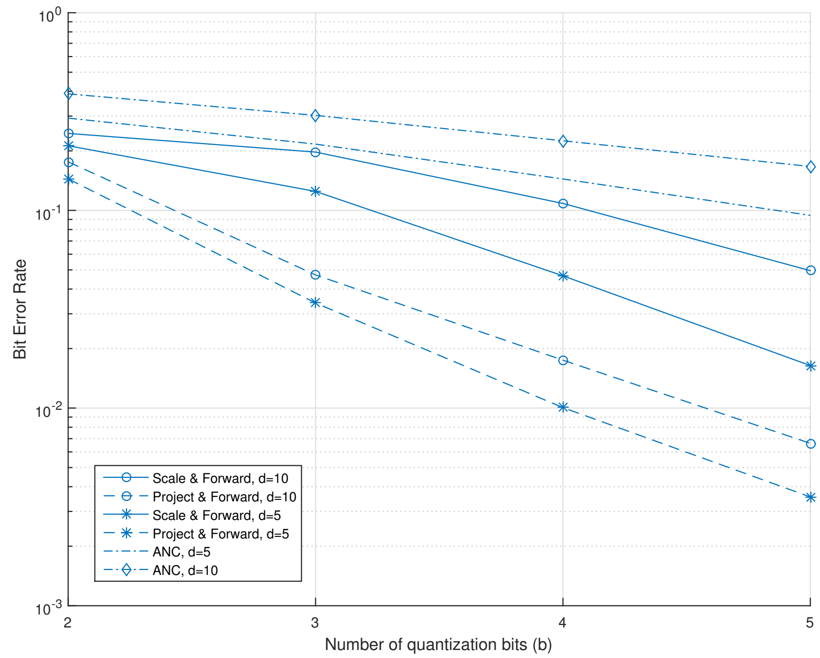


Figure 4.2: BER versus number of quantization bits for different methods in stacked butterfly network for $l = 2$.

4.5.2 Results

Simulations are done using MATLAB for the line and stacked butterfly network using codebook construction of Section 4.5.1. Local encoding vectors for each node are chosen uniformly random from $\mathcal{U}(-1, 1)$. Quantization is done on every edge after construction of output vector in its corresponding node. In the stacked butterfly network for the case of project and forward we set τ in (4.15) equal to two.

Fig. 4.2 shows the bit error rate (BER) versus the number of bits representing each real number (b) in the stacked butterfly network with different number of stages $d = 5, 10, 20$ and for a codebook of size 4, i.e., $l = 2$. Evidently, our method significantly improves the BER compared to ANC.

Using the line network, we simulate multi-resolution multicast aspect of our method in the presence of both quantization noise and additive white Gaussian

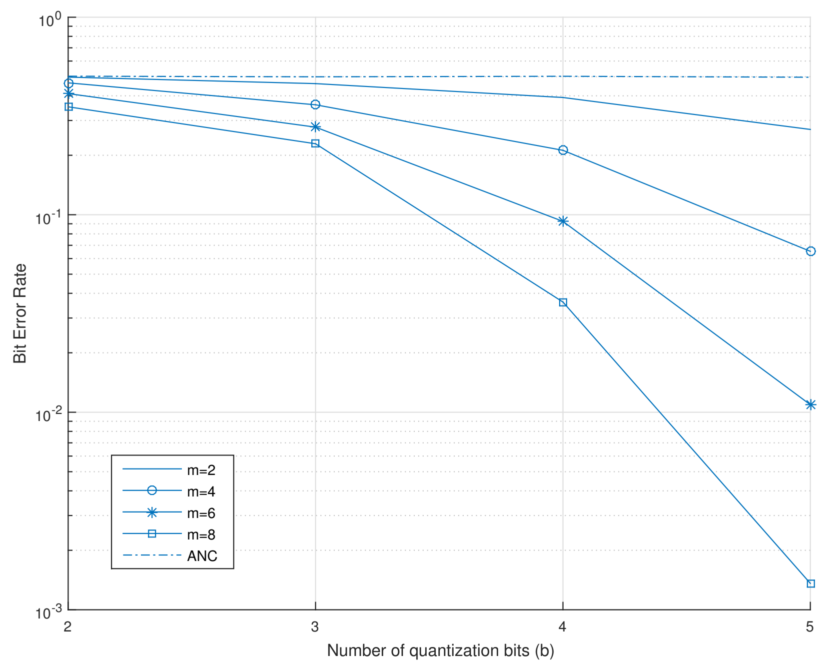


Figure 4.3: Effect of different download capacities in terms of BER in line network for $l = 2$.

noise. For this purpose, we set $d = 20$ and $\sigma^2 = 1/b$ (so that we can provide a 2D plot), where σ^2 is the variance of additive white Gaussian noise. BER for sinks with different download capacities (different m) is presented in Fig. 4.3 for $l = 2$ when intermediate nodes project and forward. As illustrated our scheme also improves the BER compared to ANC in the presence of quantization and additive white Gaussian noise.

Finally, we examine our method for different number of subspaces (2^l). Fig 4.4 shows the bit error rate for $d = 20$ and different values of l in the stacked butterfly network.

4.6 Summary

While arithmetic network coding can be a useful technique for some particular network problems such as multi-resolution multicast, a major problem with

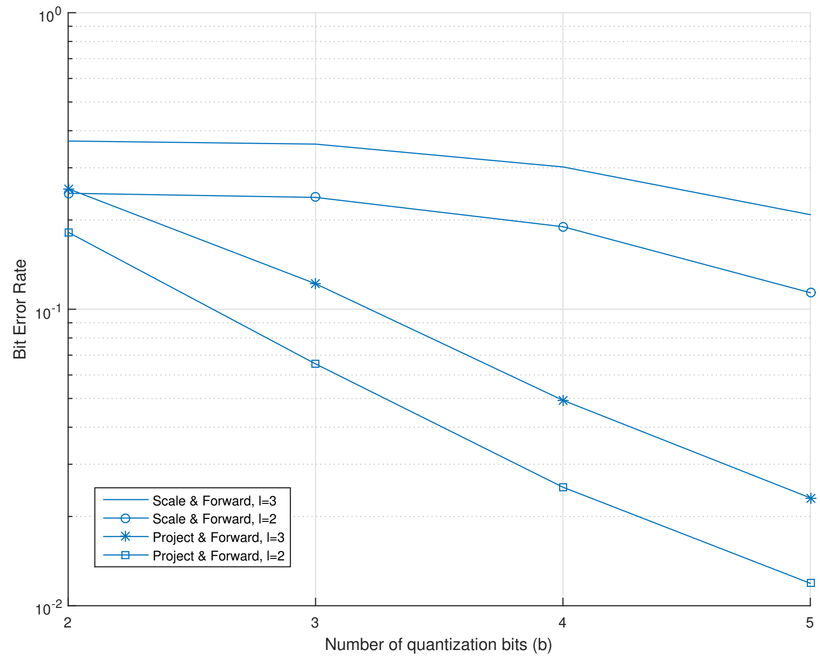


Figure 4.4: Different number of subspaces 4, 8 ($l = 2, 3$) in stacked butterfly network with $d = 20$.

ANC is noise accumulation, which can result in a lot of errors in large networks. Subspace coding is shown to have error correction abilities. In this chapter, we first discussed why subspace coding for ANC is a challenging task. Through modeling the network noise and studying its properties, we were able to suggest a framework for subspace coded ANC and its decoding. Our methods allows intermediate nodes to project and forward, resolving the noise accumulation problem. We simulated an example of subspace coded ANC and observed that our solution outperforms conventional ANC significantly.

Chapter 5

Conclusion & Future Work

5.1 Conclusion

In this thesis, we studied different problems associated with network coding and presented our solutions.

In Chapter 3 a class of networks that we call funnel networks has been studied. In this particular class of networks the incoming capacity of each intermediate node is not less than its outgoing capacity. We proved that for any general single-source non-multicast problem on funnel networks, routing achieves the max-flow min-cut bound. In other words, network coding bring no additional value in terms of throughput. The desirability of coding in funnel networks may be justified by other benefits that network coding offers. We also studied the problem of finding linear network codes for funnel networks. Since in a general non-multicast problem conventional random linear network coding fails with high probability, we proposed a method based on RLNC for finding coding coefficients in each node. Our results can be very useful in practical scenarios where designing network codes in a deterministic fashion is not feasible e.g. dynamic networks.

In Chapter 4 the problem of error accumulation and large condition numbers in an arithmetic network coding scenario is considered. Because of its powerful ability in error correction, we suggested a subspace coding framework to overcome this issue. We proposed our rank deficient decoding method for receiver(s) and a relaying approach similar to amplify and forward that we call scale and forward. We took this idea a step further and proposed another approach called project and forward in which, similar to decode and forward relaying, intermediate nodes help with error correction. Also, because in our decoding method no matrix inversion is needed, large condition numbers do not jeopardize the quality of our communication.

5.2 Future Work

5.2.1 Funnel Networks

As mentioned earlier, finding the limits of network coding is of interest. This can help us identify coding advantage and cost of using network coding in different scenarios which in turn helps us better utilize our resources and find methods to make an acceptable trade-off between coding advantages and its cost.

Examining our random linear network coding technique in different scenarios would be another interesting direction for future research. We conjecture that we can apply our technique to other types of network with some modifications. To this end, one may also look into the relationship between funnel networks and other types of networks.

5.2.2 Arithmetic Subspace Coding

To further extend this research, we suggest finding efficient code design for our subspace coding framework and decoding function. To this end, some results in the field of non-coherent MIMO transmission may be interesting. Finding similarities and differences between random linear ANC and non-coherent

MIMO can be very helpful for this purpose. As an example, in MIMO systems, characteristics of the channel is determined by many factors that cannot be controlled. In contrast, some components of the network may be changed in favor of better outcome such as the packet length and local encoding operations at intermediate nodes.

Another direction would be examining different techniques for dimension reduction at the receivers in order to combine information received from different paths in the network (diversity). Utilizing additional received packets from network is also interesting in conventional network coding methods since, in practical networks, download capacities for each sink may be different. Therefore, finding feasible methods may result in providing better quality of service.

In Chapter 4, we showed that for large dynamic networks, the noise received at each sink can be reasonably modelled by i.i.d Gaussian random variables. However, for other type of networks, this assumption may not be accurate. Examining the accuracy of this approximation and finding better models for other types of networks should be considered as well.

Bibliography

[1] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.*, 29(7):1645–1660, September 2013. ISSN 0167-739X. doi: 10.1016/j.future.2013.01.010. URL `http://dx.doi.org/10.1016/j.future.2013.01.010`.

[2] Mohamed Ali Feki, Fahim Kawsar, Mathieu Boussard, and Lieven Trappeniers. The internet of things: the next technological revolution. *Computer*, 46(2):24–25, 2013.

[3] Internet Live Stats - Internet Usage & Social Media Statistics, 2016. URL `http://www.internetlivestats.com/internet-users/`. [Online; Accessed: 27- Nov- 2016].

[4] Alcetel Lucent Bell Labs. G.W.A.T.T. (Global What if Analyzer of NeTwork Energy ConsumpTion). Bell Labs application able to measure the impact of technologies like SDN & NFV on network energy consumption. (White Paper). 2015. URL `http://gwatt.net/intro/1`.

[5] R. Ahlswede, Ning Cai, S.-Y.R. Li, and R.W. Yeung. Network information flow. *IEEE Trans. Inf. Theory*, 46(4):1204–1216, Jul. 2000. ISSN 0018-9448. doi: 10.1109/18.850663.

[6] Christina Fragouli and Emina Soljanin. Network coding fundamentals. *Foundations and Trends in Networking*, 2(1):1–133, 2007. ISSN 1554-

057X. doi: 10.1561/1300000003. URL http://dx.doi.org/10.1561/1300000003.

[7] Qiang Huo, Lingyang Song, Yonghui Li, and Bingli Jiao. Source and physical-layer network coding for correlated two-way relaying. *IET Communications*, 10(5):502–507, 2016.

[8] Makesh Pravin Wilson, Krishna Narayanan, Henry D Pfister, and Alex Sprintson. Joint physical layer coding and network coding for bidirectional relaying. *IEEE Transactions on Information Theory*, 56(11):5641–5654, 2010.

[9] Alireza Keshavarz-Haddad and Rudolf H Riedi. Bounds on the benefit of network coding for wireless multicast and unicast. *Mobile Computing, IEEE Transactions on*, 13(1):102–115, 2014.

[10] Deepak Goyal and Malay Ranjan Tripathy. Routing protocols in wireless sensor networks: A survey. In *Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on*, pages 474–480. IEEE, 2012.

[11] Kevin C Lee, Uichin Lee, and Mario Gerla. Survey of routing protocols in vehicular ad hoc networks. *Advances in vehicular ad-hoc networks: Developments and challenges*, pages 149–170, 2010.

[12] Sunil Taneja and Ashwani Kush. A survey of routing protocols in mobile ad hoc networks. *International Journal of innovation, Management and technology*, 1(3):279, 2010.

[13] S Shintre, S Katti, S Jaggi, BK Dey, D Katabi, and M Medard. ”Real” and ”Complex” network codes: Promises and challenges. In *2008 4th Workshop on Network Coding, Theory and Applications*, pages 1–6, 2008.

[14] Sachin Katti, Saurabh Shintre, Sidharth Jaggi, Dina Katabi, and Muriel Medard. Real network codes. In *Proc. Forty-Fifth Annual Allerton Conference*, pages 389–395, 2007.

[15] Ralf Koetter and Frank R Kschischang. Coding for errors and erasures in random network coding. *Information Theory, IEEE Transactions on*, 54 (8):3579–3591, 2008.

[16] Robert J McEliece. *Finite fields for computer scientists and engineers*, volume 23. Springer Science & Business Media, 2012.

[17] Karl Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10(1):96–115, 1927.

[18] Muriel Médard and Alex Sprintson. *Network coding: fundamentals and applications*. Academic Press, 2011.

[19] Samuel J Mason. Feedback theory: Further properties of signal flow graphs. 1956.

[20] S.-Y.R. Li, R.W. Yeung, and Ning Cai. Linear network coding. *IEEE Trans. Inf. Theory*, 49(2):371–381, Feb. 2003. ISSN 0018-9448. doi: 10.1109/TIT.2002.807285.

[21] Philip A. Chou, Yunnan Wu, and Kamal Jain. Practical network coding. In *Allerton Conf. on Commun., Control, and Computing*, Oct. 2003. Invited paper.

[22] R. Dougherty, C. Freiling, and K. Zeger. Linearity and solvability in multicast networks. *IEEE Trans. Inf. Theory,*, 50(10):2243–2256, Oct. 2004. ISSN 0018-9448. doi: 10.1109/TIT.2004.834751.

[23] Ami Tavory, Meir Feder, and Dana Ron. Bounds on linear codes for network multicast. *Electronic Colloq. on Computational Complexity*, 10: 033, 2003.

[24] Tracey Ho, M. Medard, R. Koetter, D.R. Karger, M. Effros, Jun Shi, and B. Leong. A random linear network coding approach to multicast. *IEEE Trans. Inf. Theory,*, 52(10):4413–4430, Oct. 2006. ISSN 0018-9448. doi: 10.1109/TIT.2006.881746.

[25] Lizhong Zheng and David N. C. Tse. Communication on the grassmann manifold: A geometric approach to the noncoherent multiple-antenna channel. *IEEE Trans. Inf. Theory*, 48(2):359–383, 2002.

[26] Thomas Honold, Michael Kiermaier, Sascha Kurz, et al. Optimal binary subspace codes of length 6, constant dimension 3 and minimum subspace distance 4. *Topics in finite fields*, 632:157–176, 2015.

[27] Daniel Heinlein, Michael Kiermaier, Sascha Kurz, and Alfred Wassermann. Tables of subspace codes. *arXiv preprint arXiv:1601.02864*, 2016.

[28] Michael Braun, Patric RJ Östergård, and Alfred Wassermann. New lower bounds for binary constant-dimension subspace codes. *Experimental Mathematics*, pages 1–5, 2016.

[29] Elisa Gorla and Alberto Ravagnani. Subspace codes from ferrers diagrams. *Journal of Algebra and Its Applications*, page 1750131, 2014.

[30] Jørgen Bang-Jensen, András Frank, and Bill Jackson. Preserving and increasing local edge-connectivity in mixed graphs. *SIAM Journal on Discrete Mathematics*, 8(2):155–178, 1995.

[31] R. Dougherty, C. Freiling, and K. Zeger. Insufficiency of linear coding in network information flow. In *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT)*, pages 264–267, Sep. 2005. doi: 10.1109/ISIT.2005.1523335.

[32] R. Dougherty, C. Freiling, and K. Zeger. Insufficiency of linear coding in network information flow. *IEEE Trans. Inf. Theory,*, 51(8):2745–2759, Aug. 2005. ISSN 0018-9448. doi: 10.1109/TIT.2005.851744.

[33] Zongpeng Li and Baochun Li. Network coding in undirected networks. CISS, 2004.

[34] Zongpeng Li, Baochun Li, and Lap Chi Lau. On achieving maximum multicast throughput in undirected networks. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2467–2485, 2006.

[35] Zongpeng Li, Baochun Li, and Lap Chi Lau. A constant bound on throughput improvement of multicast network coding in undirected networks. *Information Theory, IEEE Transactions on*, 55(3):1016–1026, 2009.

[36] M. Schwartz and M. Medard. Quasi-linear network coding. In *Network Coding (NetCod), 2014 International Symposium on*, pages 1–6, June 2014. doi: 10.1109/NETCOD.2014.6892126.

[37] Dimitri P. Bertsekas, Cover Design, Ann Gallager, C Dimitri P. Bertsekas, and Bertsekas Dimitri P. Network optimization: Continuous and discrete models, athena scientific, 1998.

[38] P. Elias, A. Feinstein, and C.E. Shannon. A note on the maximum flow through a network. *IRE Trans. Inf. Theory*, 2(4):117–119, Dec. 1956. ISSN 0096-1000. doi: 10.1109/TIT.1956.1056816.

[39] Tracey Ho and Desmond Lun. *Network Coding: An Introduction.* Cambridge University Press, New York, NY, USA, 2008. ISBN 052187310X, 9780521873109.

[40] Danilo Silva, Frank R Kschischang, and Ralf Koetter. A rank-metric approach to error control in random network coding. *Information Theory, IEEE Transactions on*, 54(9):3951–3967, 2008.

[41] Lloyd N Trefethen and David Bau III. *Numerical Linear Algebra.* SIAM, 1997.

[42] Zhiyuan Yan, Hongmei Xie, and Bruce W Suter. Rank deficient decoding of linear network coding. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 5080–5084. IEEE, 2013.

[43] D. Silva and F.R. Kschischang. Universal secure network coding via rank-metric codes. *Information Theory, IEEE Transactions on*, 57(2):1124–1135, Feb 2011. ISSN 0018-9448. doi: 10.1109/TIT.2010.2090212.

[44] Mahdy Nabaee and Fabrice Labeau. Quantized network coding for sparse messages. In *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, pages 828–831. IEEE, 2012.

[45] Zoran Utkovski and Juergen Lindner. On the construction of non-coherent space time codes from high-dimensional spherical codes. In *Spread Spectrum Techniques and Applications, 2006 IEEE Ninth International Symposium on*, pages 327–331. IEEE, 2006.

[46] Oliver Henkel. Sphere-packing bounds in the grassmann and stiefel manifolds. *IEEE Transactions on Information Theory*, 51(10):3445–3456, 2005.

[47] Ramy H Gohary and Timothy N Davidson. Noncoherent mimo communication: Grassmannian constellations and efficient detection. *IEEE Transactions on Information Theory*, 55(3):1176–1205, 2009.

[48] Ramy H Gohary and Halim Yanikomeroglu. Grassmannian signalling achieves tight bounds on the ergodic high-snr capacity of the noncoherent mimo full-duplex relay channel. *IEEE Transactions on Information Theory*, 60(5):2480–2494, 2014.