

Linear Network Codes on Cyclic and Acyclic Networks

by

Ali Esmaeili

B.Sc., Isfahan University of Technology, 2013

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

© Ali Esmaeili, 2016

University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by
photocopying or other means, without the permission of the author.

Linear Network Codes on Cyclic and Acyclic Networks

by

Ali Esmaeili

B.Sc., Isfahan University of Technology, 2013

Supervisory Committee

Dr. Aaron Gulliver, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Michael L. McGuire, Departmental Member
(Department of Electrical and Computer Engineering)

Supervisory Committee

Dr. Aaron Gulliver, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Michael L. McGuire, Departmental Member
(Department of Electrical and Computer Engineering)

ABSTRACT

Consider a network which consists of noiseless point to point channels. In this network, the source node wants to send messages to a specific set of sink nodes. If an intermediate node v has just one input channel then the received symbol by that node can be replicated and sent to the outgoing channels from v . If v has at least two incoming channels then it has two options. It can either send the received symbols one-by-one, one symbol in each time unit, or v can transmit a combination of the received symbols. The former choice takes more time compared to the latter option, which is called network coding.

In the literature, it has been shown that in a single source finite acyclic network the maximum throughput can be achieved by using linear network codes. Significant effort has been made to efficiently construct good network codes. In addition, a polynomial time algorithm for constructing a linear network code on a given network was introduced. Also an algorithm for constructing a linear multicast code on an acyclic network was introduced. Finally, a method for finding a representation matrix for the network matroid of a given network G was also introduced. This matrix can be used to construct a generic code.

In this thesis we first provide a review of some known methods for constructing linear multicast, broadcast and dispersion codes for cyclic and acyclic networks. We then give a method for normalization of a non-normal code, and also give a new algorithm for constructing a linear multicast code on a cyclic network. The construction of generic network codes is also addressed.

Contents

Supervisory Committee	ii
Abstract	iii
Table of Contents	iv
List of Tables	v
List of Figures	vi
Acknowledgements	viii
1 Introduction	1
1.1 Definitions and Concepts	3
1.1.1 Overview and Contributions	11
2 Linear Network Codes	12
2.1 Codes on Acyclic Networks	12
2.2 Codes on Cyclic Networks	16
3 Code Construction on Cyclic Networks	22
3.1 Normalization of a Non-normal Code	22
3.2 Construction of a Linear Multicast Code on a Cyclic Network	24
3.3 An Improvement of the Four Layer Network Algorithm	31
3.4 Creating a Generic Network Code on a Cyclic Network	36
4 Conclusions	38
Bibliography	40

List of Tables

Table 2.1	A list of all possible coding coefficients for Example 13.	19
Table 2.2	A simplified version of Table 2.2	20
Table 3.1	Algorithm 6 and the four layer network algorithm executed on the networks in Figures 3.5 to 3.9.	30
Table 3.2	Four sets of coding coefficients for the shuttle network obtained by Algorithm 6.	32
Table 3.3	Coding coefficients derived from the four layer network algorithm for the shuttle network.	33
Table 3.4	Coding vectors of the first code given in Table 3.2.	34
Table 3.5	Coding vectors of the code given in Table 3.3 for the shuttle network.	34
Table 3.6	Coding coefficients derived from coding vectors given in Exam- ple 19, assigned to the network shown in Figure 3.3.	37
Table 3.7	Coding coefficients related to Table 3.6 after inserting a delay in each cycle of the shuttle network. The location of D shows the position of the assigned delays.	37

List of Figures

1.1	The butterfly network with a source node and two sink nodes. . . .	2
1.2	A cyclic graph containing two cycles. The edges e_1, e_4, e_5 and e_2 form a cycle and the edges e_1, e_3 and e_2 form another cycle.	5
1.3	A simple network illustrating incoming and outgoing channels at a node.	5
1.4	A network with two imaginary channels appended at source node s	5
1.5	Two cuts $\{e_4, e_5, e_6, e_7\}$ and $\{e_1, e_2, e_3\}$ between source node s and the sink nodes $T = \{t_1, t_2\}$ with capacities 4 and 3, respectively.	6
1.6	The global encoding for the edges e_2 and e_4 is $\hat{f}_e(x) = b_1$	7
1.7	The diamond network with the imaginary channels connected to the source node.	8
1.8	A linear dispersion code on an acyclic graph.	9
2.1	Creation of a linear multicast code on an acyclic network.	14
2.2	A network whose matroid is given in Example 12.	16
2.3	A cyclic network with one cycle and its corresponding acyclic four layer network.	18
2.4	A cyclic network G and acyclic network G' derived from G	21
2.5	A generic network code on the four layer acyclic network of Figure 2.4.	21
3.1	A non-normal code (1) and a normal code derived from it (2).	23
3.2	Different components of a graph $G' = G - E'_T \setminus E_T$	26
3.3	Shuttle network with the first set of coding coefficients given in Table 3.2 obtained using Algorithm 6.	28
3.4	Shuttle network with the coding coefficients obtained by the four layer network algorithm.	29
3.5	The first cyclic network for Table 3.1.	30
3.6	The second cyclic network for Table 3.1.	31
3.7	The fourth cyclic network for Table 3.1.	31

3.8	The fifth cyclic network for Table 3.1.	32
3.9	A cyclic network G which has three single-edge components and one cycle component.	33
3.10	A cyclic network G and its associated four layer network G' . The third graph is the acyclic network $G' - \{s_4\}$	35

ACKNOWLEDGEMENTS

I would like to thank:

My supervisor, Dr. Aaron Gulliver for his guidance, patience and support. He is always open and honest in communicating with his students and I never would have completed my masters degree without his supervision. Attending grad school was one of the most challenging and best decisions I made. I am forever grateful for having him as my supervisor.

In addition to my supervisor, I would like to thank Prof. Michael L. McGuire as the supervisory committee member for my thesis, and Prof. Venkatesh Srinivasan from the Computer Science department as my external examiner for reviewing my thesis, attending my oral exam and their suggestions and tips on my research work.

Of course lastly and most important I would like to thank my family who have always been there to provide support and encouragement to me especially when I needed it the most.

*They said you wouldn't make it so far and ever since they've said it its been hard.
But never mind that night you had to cry cause you had never let it go inside. You
worked real hard and you know exactly what you want and need so believe and you
can never give up. You can reach your goals.*

Yolanda Adams

Chapter 1

Introduction

Network coding is a field of research in communication theory where a communication network is generally considered as a finite acyclic directed graph with a single source. In 2000, Ahlswede, Cai, Li, and Yeung [1] introduced the concept of network coding and network information theory. There was excitement about its possibilities and skepticism about its potential. As a simple example, consider the network shown in Figure 1.1 known as the butterfly network. In this figure, the node s is called the source node, and the nodes t_1 and t_2 are called the sink nodes. The goal is receiving messages m_1 and m_2 by both t_1 and t_2 as quickly as possible. It is evident that this is achieved if node c transmits $m_1 + m_2$ to d instead of sending m_1 and m_2 separately. The operation of getting m_1 and m_2 through node c and transmitting $m_1 + m_2$ to d is the essence of network coding.

A network can be modeled as a directed graph $G = (V, E)$ with node set V and edge set E , where the connection of nodes i and j in G is represented by an edge between these nodes. It is also assumed that there is a node in G which generates messages for transmission to other nodes. This node is called the source node. It is also assumed that there are a set of destination nodes which want to receive the messages transmitted by the source node. These nodes are called destination or sink nodes. The set of symbols may be considered as elements of a finite field F , and a message consists of ω symbols taken from F . In the butterfly network we have $\omega = 2$. The objective is to choose an arbitrary, causal mapping from input edges to output edges at each node such that each sink node in the network G can determine what was sent from the source node. The goal of network coding is to increase the speed and accuracy of message transmission from the source node to the sink nodes.

The butterfly network is a single source-two sink network. Both sinks want to

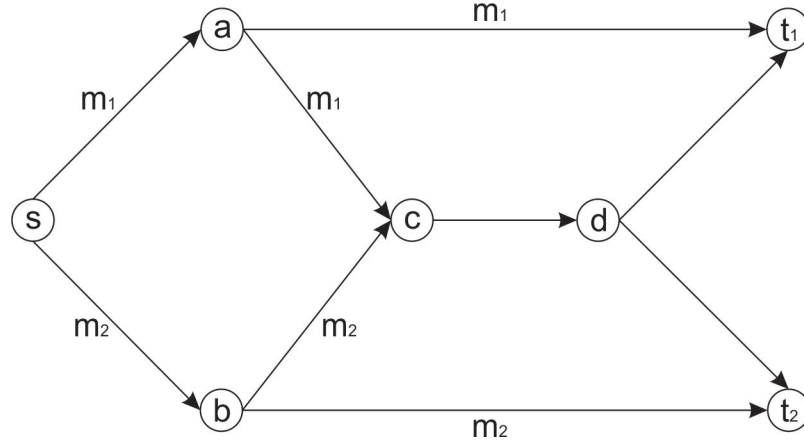


Figure 1.1: The butterfly network with a source node and two sink nodes.

know the symbols sent by the source node. In this network, the desired transmissions can be established if node c forms a new message by taking a combination of the two received messages and sends the resulting message on the channel to node d . Node d can forward the message sent by node c on the channels to the sink nodes. Sink t_1 gets message m_2 by taking a combination of message m_1 and message $m_1 + m_2$. Sink t_2 obtains message m_1 by taking a combination of message m_2 and message $m_1 + m_2$. If node c uses only routing and sends message m_1 , only sink t_2 will be able to receive both messages m_1 and m_2 , and if node c sends message m_2 then only sink t_1 will be able to receive both messages.

The butterfly network illustrates that network coding can increase the throughput of a network. There are nine message transmissions in the butterfly network when network coding is used. Without network coding, a set of nine transmissions is not able to send two messages m_1 and m_2 to the sinks t_1 and t_2 , so additional transmissions are needed.

If two nodes a and b want to send messages and the messages are sent separately then some individual channels, in this case cd , will be used more than required. If the messages are sent using linear network coding¹ the individual channels usage will be reduced and the throughput will be improved. Therefore, source separation is not an optimal solution.

In [1], the authors considered networks consisting of nodes interconnected by error free point-to-point links rather than looking at general networks where essentially

¹Linear network coding is an arbitrary mapping from the input channels to the output channels at a node in a network such that the output on an outgoing channel is a linear combination of the inputs from the incoming channels.

every link has a probabilistic effect on the message which is being sent to adjacent nodes. Thus the network model of Ahlswede et al. is a special case of the general network communication model. In [4], Li et al. showed that in a single source finite acyclic network the maximum throughput can be achieved by using linear network codes. After the work done by Li et al., significant effort has been made to efficiently construct good network codes. Jaggi and Sanders proposed a polynomial time algorithm [2] for the construction of a linear network code on a given network. It was mentioned that this random algorithm has a running time of $O(|E| * |T| * h^2)$ where h denotes the minimum cut² between the source and any sink t in the set of destination nodes T . It was also shown that this algorithm works on any finite field of size $|F| \geq 2|T|$.

The butterfly network is an example of an acyclic network, where a graph is acyclic if it has no directed cycle. Network coding on error-free cyclic networks has been studied in [3].

A linear multicast code is a linear network code such that each node in the network which has more than ω incoming edges can decode the message sent to it from the source node. This thesis proposes an efficient running time algorithm for constructing a linear multicast code on a cyclic network such that the algorithm gives a good solution in terms of the field size.

1.1 Definitions and Concepts

In the remainder of this chapter, the definitions and theorems which are needed for the following chapters are introduced below. These are required concepts regarding graph theory, cyclic and acyclic graphs, network communications, and network coding.

Definition 1. A graph G is a two-tuple $G = (V, E)$ where V is a finite set called node set, and E is a set of pairs of nodes called the edge set of G . Graphically, the vertices are denoted by points and each element $e = (a, b)$ of E is represented by a line connecting the corresponding nodes a and b .

Definition 2. A path in a graph G is a sequence $e_1 e_2 \cdots e_t$ of edges $e_i = (a_i, a_{i+1})$ where the a_i 's are (not necessarily distinct), vertices of G . The length of a path is the

²A minimum cut between two nodes p and q in a graph G is the minimum number of edges in G whose removal from G results in a graph in which p and q are not connected by a path.

number of edges in that path. Two directed paths p_1 and p_2 in G are edge-disjoint if they do not share a common edge.

Definition 3. A graph G is called a directed graph if each edge $e = (a, b)$ in E has a direction from a to b . In this case, a and b are called the tail and head of e , respectively. The set of edges whose head is a are called the incoming edges to a and the outgoing edges from a are the edges whose tail is a .

In this thesis we consider only directed graphs.

Definition 4. A closed path is a path whose first and last vertices are the same. A closed path is called a cycle. The length of a cycle is the number of edges in that cycle.

Definition 5. A single source communication network is a directed graph that has a node with no incoming edges, called the source node, and a set of nodes with no outgoing edges, called sinks or destination nodes. A network having at least one cycle is called cyclic, otherwise it is called acyclic. Each edge of a network is called a channel.

Example 1. Consider the network given in Figure 1.2. This network contains two cycles. The edges e_1 , e_3 and e_2 form a cycle of length three and edges e_1 , e_4 , e_5 and e_2 form a second cycle of length four. Note that this graph is not considered as a communication network as it has neither a source nor a sink node. The butterfly graph is an acyclic communication network with two sink nodes.

The source node where the messages are generated is denoted by s . The incoming and outgoing channels at node i are denoted by $In(i)$ and $Out(i)$, respectively. For completeness, let $In(s)$ be a set of ω imaginary channels that terminate at node s but have no originating nodes. It is assumed that parallel channels between a pair of nodes is allowed and all the channels in the network have unit capacity.

Example 2. Consider the network given in Figure 1.3. In this network we have $In(u) = \{e_1, e_2\}$ and $Out(u) = \{e_3\}$.

Example 3. In the network given in Figure 1.4, two imaginary channels are appended at source node s .

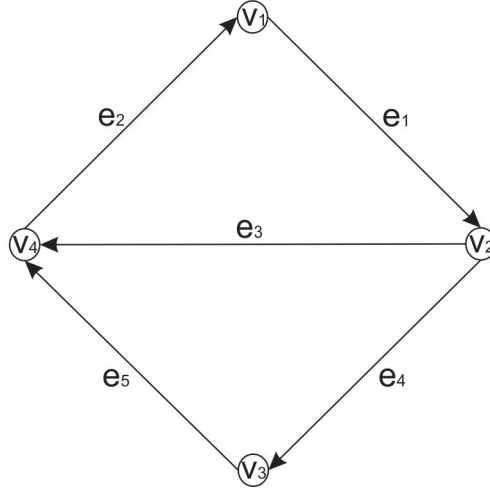


Figure 1.2: A cyclic graph containing two cycles. The edges e_1 , e_4 , e_5 and e_2 form a cycle and the edges e_1 , e_3 and e_2 form another cycle.

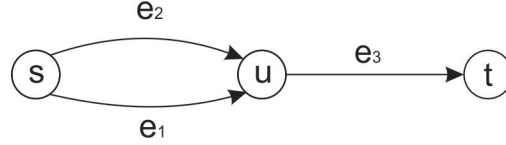


Figure 1.3: A simple network illustrating incoming and outgoing channels at a node.

Definition 6. The capacity of a channel is the maximum number of messages which can be reliably transmitted over that channel per unit time. The capacity of a cut between two nodes a and b is the sum of the capacities of the channels in that cut. The minimum cut, or simply min-cut, between two nodes a and b is a cut between these nodes that has minimum capacity among all cuts between a and b .

Example 4. Consider the network given in Figure 1.5. The edges $\{e_4, e_5, e_6, e_7\}$ form a cut between s and $T = \{t_1, t_2\}$. If each edge has a capacity of one message per unit time then the capacity of this cut is four. The set $\{e_1, e_2, e_3\}$ is a min-cut between s

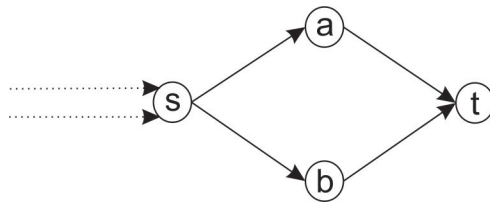


Figure 1.4: A network with two imaginary channels appended at source node s .

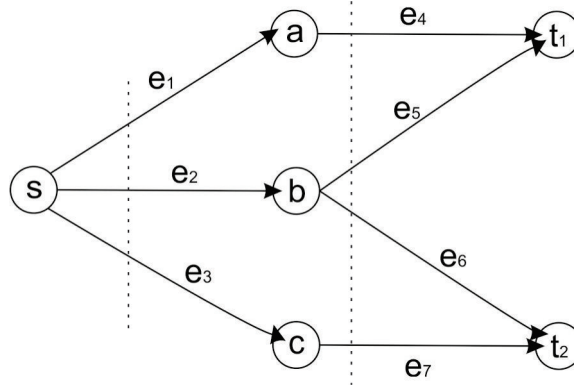


Figure 1.5: Two cuts $\{e_4, e_5, e_6, e_7\}$ and $\{e_1, e_2, e_3\}$ between source node s and the sink nodes $T = \{t_1, t_2\}$ with capacities 4 and 3, respectively.

and $T = \{t_1, t_2\}$.

Definition 7. The maximum flow, denoted max-flow, between a source node s and a sink t in a network is a data flow of maximum value that can be transmitted from s to t .

Theorem 1. [9] Let s and t be two nodes in a network. The maximum flow between these nodes is equal to the min-cut capacity between the nodes.

Based on this theorem, if in a network β messages can be sent from the source node to each sink node, then β is less than or equal to the minimum of the max-flow of all sinks in the network.

Now we give some elementary concepts of network coding theory.

Definition 8. [9] A local encoding for an ω -dimensional network code on an acyclic network over a field F is defined as a mapping $\hat{k}_e : F^{|In(t)|} \rightarrow F$ for every channel e and every node t in the network such that $e \in Out(t)$.

Definition 9. [9] A global encoding for an ω -dimensional network code on an acyclic network over a field F is a mapping $\hat{f}_e : F^\omega \rightarrow F$ for every channel e in the network such that $\hat{f}_e(x)$ is uniquely determined by the input edges to e via the local encoding mapping \hat{k}_e .

Example 5. Consider the network given in Figure 1.6. We have $\hat{f}_e(x) = b_1$ for $e \in \{e_2, e_4\}$ and $\hat{f}_e(x) = b_2$ for $e \in \{e_1, e_3\}$. Also $\hat{k}_e(b_1, b_2) = b_2$ for $e \in \{e_1, e_3\}$ and $\hat{k}_e(b_1, b_2) = b_1$ for $e \in \{e_2, e_4\}$.

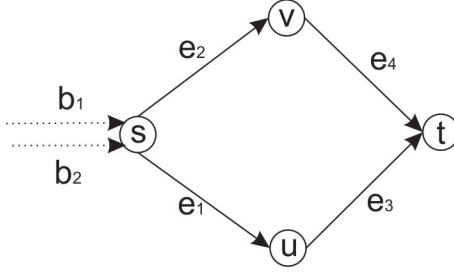


Figure 1.6: The global encoding for the edges e_2 and e_4 is $\hat{f}_e(x) = b_1$.

A local encoding mapping can be represented by a matrix $K_t = (k_{d,e})$ where $k_{d,e}$ is the local encoding kernel for every adjacent pair of edges (d, e) in the network. For each non-sink node we define K_t by

$$K_t := (k_{d,e})_{d \in \text{In}(t), e \in \text{Out}(t)}. \quad (1.1)$$

A network code $K_t = (k_{d,e})$ is called linear if to each edge e an ω -dimensional column vector f_e can be assigned such that these vectors for the imaginary incoming edges to s form the standard basis for F^ω and $f_e = \sum_{d \in \text{In}(t)} k_{d,e} f_d$ for each $e \in \text{Out}(t)$. The vector f_e is called the global encoding kernel for channel e . Then given an ω -dimensional message x , the symbol transmitted on a channel e can be computed using

$$\hat{f}_e(x) = x f_e, \quad (1.2)$$

where by $x f_e$ we mean the matrix product of $1 \times \omega$ matrix x and the $\omega \times 1$ matrix f_e . A code C is called normal if it gives a unique set of coding vectors.

Example 6. Consider the diamond network given in Figure 1.7. Assuming that $K_s = \begin{pmatrix} w & x \\ y & z \end{pmatrix}$, $K_a = u$, $K_b = v$ and $K_c = \begin{pmatrix} p \\ q \end{pmatrix}$, we get

$$f_{c,t} = p \begin{pmatrix} uw \\ uy \end{pmatrix} + q \begin{pmatrix} vx \\ vz \end{pmatrix} = \begin{pmatrix} puw+qvx \\ puy+qvz \end{pmatrix}. \quad (1.3)$$

For any node set T of a given network G and a linear network code on G , we denote by V_T the vector space generated by the global coding vectors f_e associated to the incoming edges to the nodes in T .

Theorem 2. [9] [Max-Flow Bound for Linear Network Coding] For any collection T of non-source nodes, an ω -dimensional linear network code on an acyclic network

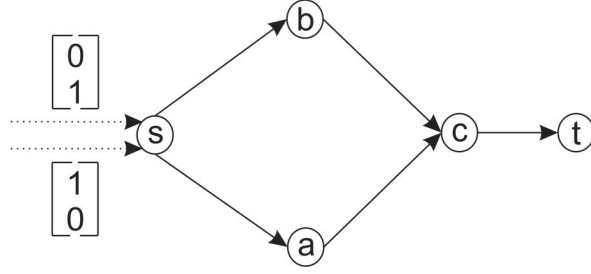


Figure 1.7: The diamond network with the imaginary channels connected to the source node.

satisfies the following property

$$\dim(V_T) \leq \min\{\omega, \max flow(T)\}. \quad (1.4)$$

Theorem 3. [4] For any collection of channels $\xi \subset E$ of an acyclic network G , an ω -dimensional linear network code on G satisfies the following property

$$\dim(V_\xi) \leq \min\{\omega, \max flow(\xi)\}. \quad (1.5)$$

Definition 10. An ω -dimensional linear network code on an acyclic network is called a linear multicast code if

$$\dim(V_t) = \omega \text{ for every non-source node } t \text{ with } \max flow(t) \geq \omega.$$

Thus, a linear network code is a linear multicast code if for every non-source node t with ω or more incoming edges, the dimension of the vector space generated by the vectors of the incoming edges is equal to ω .

Definition 11. An ω -dimensional linear network code on an acyclic network is called a linear broadcast code if

$$\dim(V_t) = \min\{\omega, \max flow(t)\} \text{ for every non-source node } t.$$

Thus, a linear network code is a linear broadcast code if for every non-source node t , the dimension of the vector space generated by the vectors of the incoming edges is equal to the minimum of ω and the number of incoming edges to t .

Definition 12. An ω -dimensional linear network code on an acyclic network is called a linear dispersion code if $\dim(V_T) = \min\{\omega, \max flow(T)\}$ for every collection T of

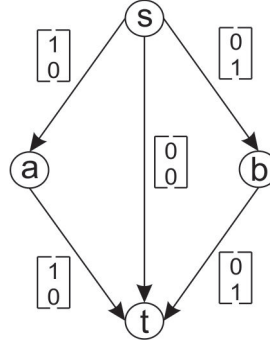


Figure 1.8: A linear dispersion code on an acyclic graph.

non-source nodes. Thus, a linear network code is a linear dispersion code if for every set of non-source nodes T , the dimension of the vector space generated by the vectors of the incoming edges is equal to the minimum of ω and the number of incoming edges to T .

A transmitted message x from the source node s consisting of ω symbols can be uniquely determined at a node t if and only if $\dim(V_t) = \omega$.

Theorem 4. [4] A node t in a linear multicast or linear broadcast code can recover a transmitted message x from the source node if and only if $\maxflow(t) \geq \omega$.

Theorem 5. [4] A collection T of non-source nodes in a linear dispersion code can recover a transmitted message x if and only if $\maxflow(T) \geq \omega$.

According to the given definitions, it is easy to verify that every linear dispersion code is a linear broadcast and every linear broadcast code is a linear multicast. However, the converse is not necessarily true, that is a linear broadcast code is not necessarily a linear dispersion code and a linear multicast code may or may not be linear broadcast. Also, in general a linear code may or may not be a multicast code.

Example 7. Consider the network given in Figure 1.8. In this network we have $\omega = 2$, $\dim(t) = 2$, $\dim(a) = 1$, $\dim(b) = 1$, $\dim(a, b) = 2$, $\dim(a, t) = 2$, $\dim(b, t) = 2$, $\dim(a, b, t) = 2$, $\maxflow(a, b) = 2$, $\maxflow(a, t) = 3$, $\maxflow(b, t) = 3$, $\maxflow(a, b, t) = 3$, $\maxflow(t) = 3$.

Based on the above definitions, this network code is a linear multicast, broadcast and dispersion code.

Theorem 6. [4] There exists an ω -dimensional linear dispersion code, and hence a linear broadcast and linear multicast code, on an acyclic network for any sufficiently large base field F .

Theorem 7. [4] An ω -dimensional linear network code for an acyclic network can be constructed by choosing the local encoding kernels $k_{(d,e)}$ for all adjacent pairs of channels (d,e) independently according to the uniform distribution on the field F , with success probability tending to 1 as $|F| \rightarrow \infty$.

At the end of this section we give some definitions on matroid theory. It has been shown that there is a close relationship between this theory and linear network coding theory.

Definition 13. [5] A matroid M on a finite set S is a finite collection of subsets of S that satisfies the following three axioms:

The empty set is in M .

If a set X is an element of M , then any subset of X is also in M .

If X and Y are in M and $|X| > |Y|$, then there is an element $x \in X - Y$ such that $Y \cup \{x\}$ is in M . This axiom is also known as the exchange property.

Example 8. Consider the set $A = \{a, b, c, d, e\}$. The set $M = \{\{\}, \{a\}, \{b\}, \{d\}, \{e\}, \{a, b\}, \{d, e\}\}$ is not a matroid for the set A because $\{a, b\}$ and $\{d\}$ are in M but neither $\{a, d\}$ nor $\{b, d\}$ is in M .

Definition 14. [5] Given a matroid M on a set S , any subset of S that is in M is called an independent set, otherwise it is a dependent set. An independent set is called a basis if it is not a proper subset of another independent set. In other words, any element of M with maximum size is called a basis for the matroid.

Example 9. Consider the set $A = \{a, b, c, d, e\}$ and the matroid $M = \{\{\}, \{b\}, \{c\}, \{d\}, \{e\}, \{b, c\}, \{b, d\}, \{b, e\}, \{c, d\}, \{d, e\}, \{b, c, d\}, \{b, d, e\}\}$. The bases for M are $\{b, d, e\}$ and $\{b, c, d\}$ which have cardinality three.

Definition 15. A dependent set B is called a circuit if every proper subset of B is independent.

Definition 16. [Network matroid] Given a network G , we consider a set B of edges in G if and only if there are some edge-disjoint paths connecting the source node s to the elements of B . It is known that the collection of such sets B form a matroid called *the network matroid* of G .

Definition 17. [3] [Matroid of a linear network code] Given a linear network code C on a network G , the matroid (E, I_C) , in which E is the edge set of G and I_C is the family of subsets of edges whose coding vectors are linearly independent, is called the induced matroid of the linear network code C .

The following definition is useful when cyclic networks and network coding on cyclic networks is studied.

Definition 18. A delay function on a network G is a nonnegative integer function t , defined over the set of adjacent pairs, such that on every cycle there is at least one edge (d, e) for which we have $t(d, e) > 0$. A linear network code on a cyclic network is said to be t -causal if the coding coefficient for every adjacent pair (d, e) is divisible by $D^{t(d, e)}$ where D stands for the delay in receiving data by an edge. This notation originates from the concepts of convolutional coding theory in classical coding.

1.1.1 Overview and Contributions

In this chapter some of the basic concepts were introduced. The main problem is given in the second chapter. The focus of the thesis will switch towards the contributions in the third chapter and in the last chapter some conclusions will be provided. In the next chapter some algorithms for constructing linear multicast, linear broadcast and linear dispersion codes on a network are introduced. These algorithms are the most common algorithms in the literature.

In a general network the local encoding kernels may not introduce a set of global coding vectors or give more than one set of solutions. In this case, the code is considered non-normal. In Chapter 3 a method for normalization of a non-normal code is provided. A new algorithm for creating multicast codes on cyclic networks is also given such that a four layer acyclic network is no longer needed. Based on a theorem in Chapter 3, a new approach for creating generic network codes on cyclic networks is provided. Finally a theorem is provided which improves the four layer network algorithm.

Chapter 2

Linear Network Codes

In the first chapter, we introduced the definitions required for network coding. In this chapter we study some of the previously introduced methods, such as those in [3, 8], for creating network codes such as linear multicast, linear broadcast and linear dispersion codes on cyclic and acyclic networks. Yang, Ming and Huang [8] proposed an algorithm for constructing a linear multicast code on an acyclic network which is given in Algorithm 1. This algorithm is also known as the network matroid algorithm.

2.1 Codes on Acyclic Networks

Algorithm 1. [8] Given a network G with n edges and ω imaginary input edges to the source node s , the following process determines a multicast code on G .

- Set the size of the global encoding kernel matrix F to $\omega * n$.
- Set the size of the local encoding kernel matrix K to $n * n$.
- Set the size of the source message matrix U to $\omega * n$.
- Label all the edges (except the imaginary edges) with integrals.
- Find the multicast nodes in the network.
- Get the matrices K and U directly from the topology of the network.
- For each multicast node t take out the vectors from matrix F that correspond to the edges entering that multicast node and align these vectors such that they form a new $\omega \times \omega$ matrix H_t .

- Assign non-zero numbers in the matrix K such that the matrix H_t for each multicast node t is full rank.

The relation between the global encoding kernel and the local encoding kernel can be obtained by the formula $F = U * [I - K]^{-1}$ [7].

Example 10. Consider the network given in Figure 2.1. We have $\omega = 3$ and the multicast node is y . From the structure of the network we have

$$\begin{aligned}
 k = & \begin{pmatrix} 0 & 0 & 0 & k_{1,4} & k_{1,5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{2,6} & k_{2,7} & k_{2,8} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{3,9} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{4,10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{5,11} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{6,10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{7,11} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{8,12} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{9,12} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
 k^2 = & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{1,4} * k_{4,10} & k_{1,5} * k_{5,11} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{2,6} * k_{6,10} & k_{2,7} * k_{7,11} & k_{2,8} * k_{8,12} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{3,9} * k_{9,12} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
 F = U * [I - K]^{-1} = U * [I + K + K^2] = & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} * \\
 & \begin{pmatrix} 1 & 0 & 0 & k_{1,4} & k_{1,5} & 0 & 0 & 0 & 0 & k_{1,4} * k_{4,10} & k_{1,5} * k_{5,11} & 0 \\ 0 & 1 & 0 & 0 & 0 & k_{2,6} & k_{2,7} & k_{2,8} & 0 & k_{2,6} * k_{6,10} & k_{2,7} * k_{7,11} & k_{2,8} * k_{8,12} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & k_{3,9} & 0 & 0 & k_{3,9} * k_{9,12} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & k_{4,10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & k_{5,11} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & k_{6,10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & k_{7,11} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & k_{8,12} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & k_{9,12} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \\
 & \begin{pmatrix} 1 & 0 & 0 & k_{1,4} & k_{1,5} & 0 & 0 & 0 & 0 & k_{1,4} * k_{4,10} & k_{1,5} * k_{5,11} & 0 \\ 0 & 1 & 0 & 0 & 0 & k_{2,6} & k_{2,7} & k_{2,8} & 0 & k_{2,6} * k_{6,10} & k_{2,7} * k_{7,11} & k_{2,8} * k_{8,12} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & k_{3,9} & 0 & 0 & k_{3,9} * k_{9,12} \end{pmatrix}
 \end{aligned}$$

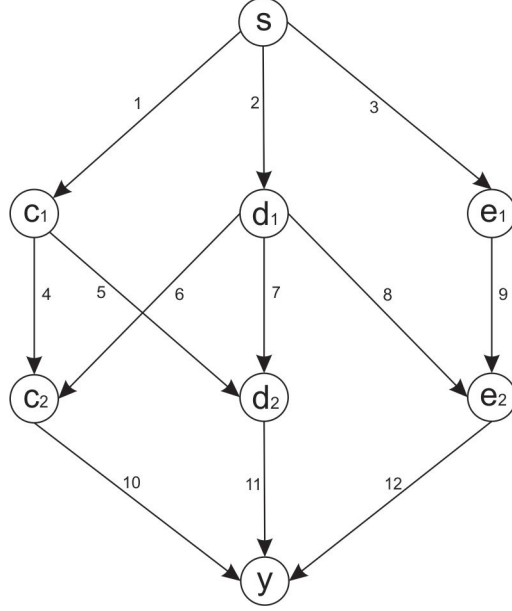


Figure 2.1: Creation of a linear multicast code on an acyclic network.

Edges 10, 11 and 12 are connected to y . Therefore, we consider columns 10, 11 and 12 in matrix F and put them together to get the following matrix H .

$$H = \begin{pmatrix} k_{1,4} * k_{4,10} & k_{1,5} * k_{5,11} & 0 \\ k_{2,6} * k_{6,10} & k_{2,7} * k_{7,11} & k_{2,8} * k_{8,12} \\ 0 & 0 & k_{3,9} * k_{9,12} \end{pmatrix}$$

The matrix H has full rank if and only if it has non-zero determinant. Hence we set

$$|H| = k_{3,9} * k_{9,12} * (k_{1,4} * k_{4,10} * k_{2,7} * k_{7,11} - k_{1,5} * k_{5,11} * k_{2,6} * k_{6,10}) \neq 0. \quad (2.1)$$

One of the possible answers in F_2 for this relation is as follows:

$$\begin{aligned} k_{3,9} &= k_{9,12} = k_{1,4} = k_{4,10} = k_{2,7} = k_{7,11} \\ &= k_{1,5} = k_{5,11} = k_{2,6} = 1, \text{ and } k_{6,10} = 0. \end{aligned} \quad (2.2)$$

In order to create a linear broadcast code the procedure is the same as creating a linear multicast code; however we should consider all broadcast nodes rather than only multicast nodes. Also we must be sure that the matrix of any node which has x input edges with $x < \omega$ has a rank of x .

Example 11. Consider the network given in Figure 2.1. The matrix F is given as follows:

$$F = \begin{pmatrix} 1 & 0 & 0 & k_{1,4} & k_{1,5} & 0 & 0 & 0 & 0 & k_{1,4} * k_{4,10} & k_{1,5} * k_{5,11} & 0 \\ 0 & 1 & 0 & 0 & 0 & k_{2,6} & k_{2,7} & k_{2,8} & 0 & k_{2,6} * k_{6,10} & k_{2,7} * k_{7,11} & k_{2,8} * k_{8,12} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & k_{3,9} & 0 & 0 & k_{3,9} * k_{9,12} \end{pmatrix}$$

The related four H_i matrices are given below.

$$H_1 = \begin{pmatrix} k_{1,4} & 0 \\ 0 & k_{2,6} \\ 0 & 0 \end{pmatrix} \quad H_2 = \begin{pmatrix} k_{1,5} & 0 \\ 0 & k_{2,7} \\ 0 & 0 \end{pmatrix} \quad H_3 = \begin{pmatrix} 0 & 0 \\ k_{2,8} & 0 \\ 0 & k_{3,9} \end{pmatrix}$$

$$H_4 = \begin{pmatrix} k_{1,4} * k_{4,10} & k_{1,5} * k_{5,11} & 0 \\ k_{2,6} * k_{6,10} & k_{2,7} * k_{7,11} & k_{2,8} * k_{8,12} \\ 0 & 0 & k_{3,9} * k_{9,12} \end{pmatrix}$$

We need H_1 , H_2 , and H_3 to be of rank two and H_4 of rank three. Therefore, a possible answer in F_2 is as follows.

$$\begin{aligned} k_{1,4} &= k_{1,5} = k_{2,6} = k_{2,7} = k_{2,8} = k_{3,9} = \\ k_{9,12} &= k_{4,10} = k_{7,11} = k_{6,10} = 1, \text{ and } k_{5,11} = 0. \end{aligned} \tag{2.3}$$

The procedure of generating a linear dispersion code is the same with the only difference that we need to consider all possible collection of nodes rather than only multicast nodes; thus construction of a linear dispersion code is more complicated than a linear multicast code or a linear broadcast code.

In order to create a generic network code each set of the elements in the network matroid needs to be linearly independent. To extract the network matroid from the network topology we can use Algorithm 2.

Algorithm 2. [8] Given a network G , the following process determines the associated network matroid.

- Label all of the non-imaginary edges in the network.
- Find all the paths from the source node to the sink nodes.
- Select any ω edges $a_1, a_2, \dots, a_\omega$. If there exists ω edge disjoint paths $m_1, m_2, \dots, m_\omega$ such that a_i is in m_i , for $1 \leq i \leq \omega$, then put that set of ω edges and its subsets into the network matroid.

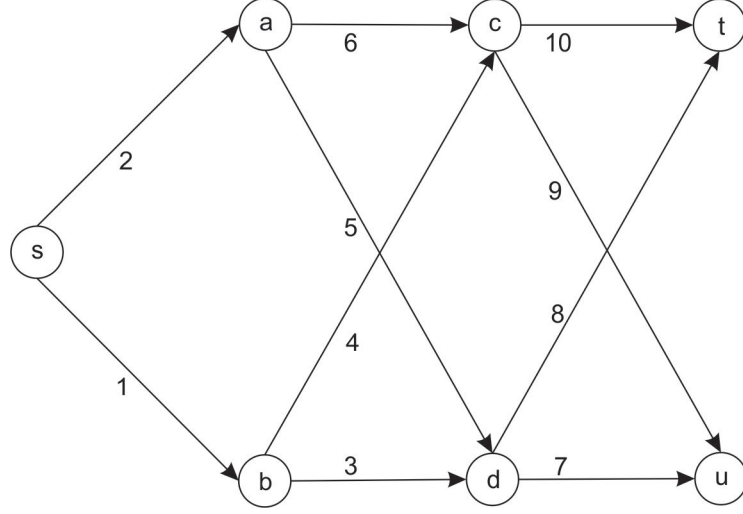


Figure 2.2: A network whose matroid is given in Example 12.

Example 12. Consider the network given in Figure 2.2. The corresponding network matroid formed by Algorithm 2 is given below.

$$M = \{\{\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}, \{9\}, \{10\}, \{1, 2\}, \{1, 5\}, \{1, 6\}, \{1, 7\}, \{1, 8\}, \{1, 9\}, \{1, 10\}, \{2, 3\}, \{2, 4\}, \{2, 7\}, \{2, 8\}, \{2, 9\}, \{2, 10\}, \{3, 5\}, \{3, 6\}, \{3, 7\}, \{3, 8\}, \{3, 9\}, \{3, 10\}, \{4, 5\}, \{4, 6\}, \{4, 7\}, \{4, 8\}, \{4, 9\}, \{4, 10\}, \{5, 7\}, \{5, 8\}, \{5, 9\}, \{5, 10\}, \{6, 7\}, \{6, 8\}, \{6, 9\}, \{6, 10\}, \{7, 8\}, \{7, 9\}, \{7, 10\}, \{8, 9\}, \{8, 10\}, \{9, 10\}\}.$$

2.2 Codes on Cyclic Networks

The algorithm in [3] can be used to create a linear multicast code on a cyclic network. In this algorithm, to a given cyclic network G an acyclic network G' is associated and it is shown that under a constraint any multicast code on G' gives a multicast code on G . In the following we first explain the method of deriving G' from G and then consider code construction for G given in Algorithm 4.

Given a cyclic network G , the associated acyclic network G' is created using Algorithm 3. Algorithm 3 is also known as the four layer network algorithm.

Algorithm 3. Let G be a given network with edge set E and source node s .

- Enter the source node for the acyclic network.
- Corresponding to each edge $e \in E \setminus \text{Out}(s)$, consider a node e_1 and add an edge $e_{(1)}$ from s to e_1 . These edges form the first layer of the four-layer acyclic

network G' .

- Corresponding to each edge $e \in E$, create a node e_2 .
- For each adjacent pair (d, e) in G , install an edge denoted \overline{de} from e_1 to d_2 ; and when $e \in E \setminus \text{Out}(s)$ install an edge $e_{(2)}$ from e_1 to e_2 . These edges form the second layer of G' .
- Corresponding to each edge $e \in E$, consider a node e_3 as well as an edge $e_{(3)}$ from e_2 to e_3 . This step constructs the third layer of G' .
- Add a node v_4 corresponding to each node $v \in G$ that is either the source node or a sink node.
- Corresponding to each edge $e \in E \setminus \text{Out}(s)$ install an edge from e_3 to s_4 .
- Arbitrarily take ω edge-disjoint paths in G that start from s and end at v . For every $e \in E$, install an edge from e_3 to v_4 unless $e \in \text{In}(v)$ and e is an edge on these paths.

Four-layer code constraint We say a given code on G' satisfies code-constraint if the coding coefficient is either 0 or 1 for every adjacent pair in the form of $(e_{(1)}, x)$, and that the coding coefficient is 1 for every adjacent pair that is in the form of $(e_{(1)}, e_{(2)})$ or $(e_{(2)}, e_{(3)})$. By a code on G' we mean a code satisfying this constraint.

The following algorithm gives a code construction method for a given cyclic network G by making use of a code for the corresponding network G' .

Algorithm 4. Let G be a given cyclic network.

- Create the acyclic network G' corresponding to the cyclic network G .
- Construct a linear multicast code on the acyclic network G' using any known method (such as the Jaggi-Sanders method or the network matroid algorithm).
- For each adjacent pair (d, e) in the cyclic network G , use the coefficient $k_{d,e} = -k'_{\overline{de}, d(3)}$ between \overline{de} and $d(3)$ in the acyclic network G' .

Example 13. Consider the cyclic network given in Figure 2.3. The right figure is the related acyclic network. One of the possible choices for the coding vectors is given in the third component of Figure 2.3 that have been found based on the following

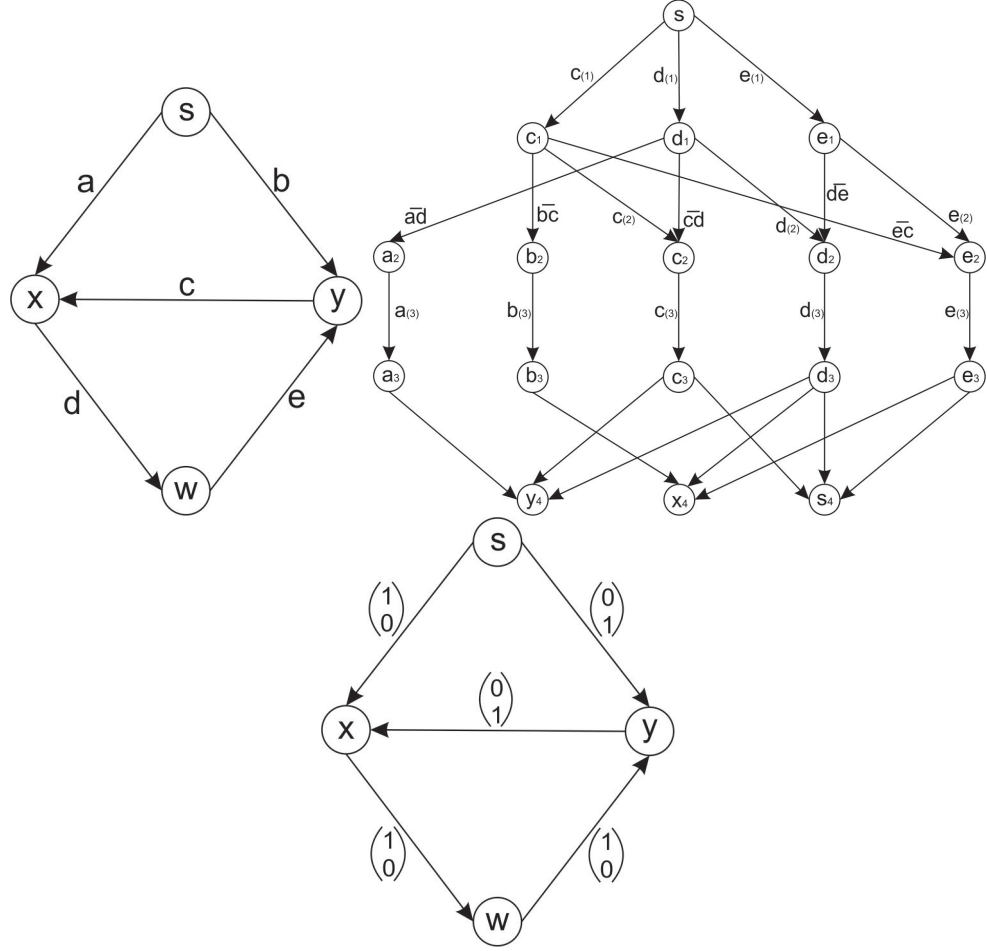


Figure 2.3: A cyclic network with one cycle and its corresponding acyclic four layer network.

coding coefficients:

We set $k_{\bar{c}d, c(3)} = k_{\bar{e}c, e(3)} = 0$ and get $k_{c,d} = k_{e,c} = 0$,
and by setting $k_{\bar{b}c, b(3)} = k_{\bar{a}d, a(3)} = k_{\bar{d}e, d(3)} = 1$ we get $k_{b,c} = k_{a,d} = k_{d,e} = 1$.

A list of all possible options for the coding coefficients for Example 13 is given in Table 2.1.

Theorem 8. [3] Let C' be an F -linear multicast code on G' . Given a delay function t on G , let C_t denote the $F[(D)]$ -linear network code prescribed by the coding coefficients

$$k_{d,e} = -D^{t(d,e)} k'_{\bar{d}e, d_3}.$$

Then C_t is a t -causal $F[(D)]$ -linear multicast code on G .

Table 2.1: A list of all possible coding coefficients for Example 13.

$K_{c_{(1)}, \bar{e}\bar{c}}$	$K_{\bar{e}\bar{c}, e_{(3)}}$	$K_{d_{(1)}, \bar{c}\bar{d}}$	$K_{\bar{c}\bar{d}, c_{(3)}}$	$K_{e, c}$	$K_{c, d}$
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	1
0	1	1	0	1	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	1	1
1	1	1	0	1	0

Therefore, considering Theorem 8 and Definition 18, we must add at least one delay to each each loop in the network G to have a casual multicast code on G . Hence, Table 2.1 can be simplified to Table 2.2.

In the rest of this section we consider generic codes on cyclic networks. Let G be a given cyclic network. An acyclic network G' , described below, is assigned to G . Using this acyclic network G' , the authors of [6] have proposed a method for finding a representation matrix for the network matroid of G . This matrix can be used for constructing a generic code. The given approach is summarized in the following algorithm.

Let G be a given cyclic network with source node s , $\deg(s) = \omega$, and edge set E . Construct a bipartite graph $G_b = (S \cup T, E')$ where the set of nodes S is $S = \{e_1, \dots, e_{|E|}\}$ and the set of vertices T is a copy of edges in $E \setminus \text{Out}(s)$ denoted $T = \{\hat{e}_{\omega+1}, \dots, \hat{e}_{|E|}\}$; the set of edges E' consists of all (e_i, \hat{e}_i) , $\omega < i \leq |E|$, and all (e_i, \hat{e}_j) such that $e_i e_j$ is a path of length 2 in G .

Algorithm 5. Consider a given cyclic network G and the associated bipartite graph $G_b = (S \cup T, E')$.

- Add a node s and a node u_3 to G_b ; connect s to each node in T and connect all

Table 2.2: A simplified version of Table 2.2

$K_{e,c}$	$K_{c,d}$	$K_{d,e}$	f_a	f_c	f_b	f_e
0	0	1	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$
0	0	D	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ D \end{pmatrix}$
0	1	1	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$
0	1	D	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} D \\ D \end{pmatrix}$
0	D	1	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ D \end{pmatrix}$
1	1	D	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} \frac{D}{1+D} \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} \frac{D}{1+D} \\ 0 \end{pmatrix}$
1	D	1	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} \frac{1}{1+D} \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} \frac{1}{1+D} \\ 0 \end{pmatrix}$
D	1	1	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} \frac{1}{1+D} \\ \frac{1}{1+D} \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} \frac{D}{1+D} \\ \frac{1}{1+D} \end{pmatrix}$

the nodes in S to u_3 . Denote the obtained network by G' .

- Construct an $(|E| - \omega)$ -dimensional generic network code on the constructed acyclic network G' .
- Form a matrix M by juxtaposing the coding vectors f_e such that $\text{tail}(e) \in S$.
- Perform row operations on M to obtain M in the form

$$\left(\begin{array}{ccc|ccc} e_1 & \dots & e_\omega & e_{\omega+1} & \dots & e_{|E|} \\ & & D & & & I_{|E|-\omega} \end{array} \right) \quad (2.4)$$

- Using (2.4), construct the matrix

$$\left(\begin{array}{ccc|ccc} e_1 & \dots & e_\omega & e_{\omega+1} & \dots & e_{|E|} \\ & & I_\omega & & & -D^T \end{array} \right) \quad (2.5)$$

- The final result will be a matrix representation for the network matroid of G .

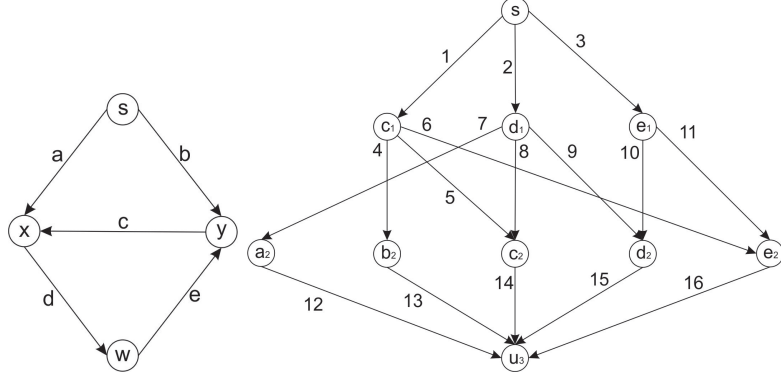


Figure 2.4: A cyclic network G and acyclic network G' derived from G .

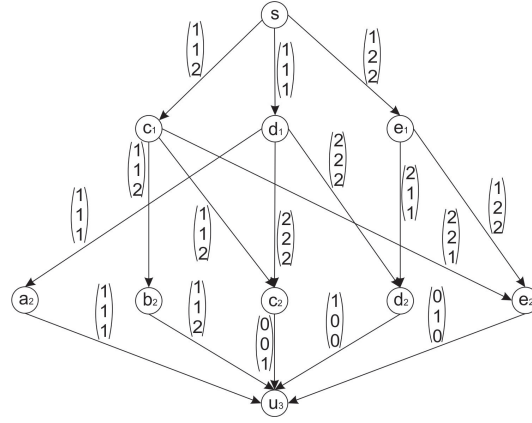


Figure 2.5: A generic network code on the four layer acyclic network of Figure 2.4.

Example 14. Consider the cyclic network and its associated acyclic network given in Figure 2.4. The generic network code of the acyclic network is given in Figure 2.5. The matrix consisting of coding vectors of edges whose tail is in S is given by (2.6). The matrix given by (2.7) represents the network matroid of the original cyclic network.

$$\begin{pmatrix} e_1 & e_2 & e_3 & e_4 & e_5 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 2 & 0 & 0 & 1 \end{pmatrix} \quad (2.6)$$

$$\begin{pmatrix} e_1 & e_2 & e_3 & e_4 & e_5 \\ 1 & 0 & 2 & 2 & 2 \\ 0 & 1 & 2 & 2 & 1 \end{pmatrix} \quad (2.7)$$

Chapter 3

Code Construction on Cyclic Networks

In the previous chapter we discussed some algorithms that are used for creating linear multicast, linear broadcast and linear dispersion codes on cyclic networks. In this chapter we provide a new approach for constructing a linear network code on a cyclic network. A set of local encoding kernels $k_{d,e}$ may not introduce a set of global coding vectors or give more than one set of solutions. A network code C with $K_C = (k_{d,e})$ is called normal if it gives a unique set of global coding vectors. Due to this, in the first part of this chapter we consider deriving a normal code from a randomly constructed non-normal code. In the second part of this chapter we introduce our proposed algorithm for creating a linear multicast code for a given cyclic network. In the third part we will switch our focus to the four layer network and provide a theorem which can be used to improve the four layer network algorithm.

3.1 Normalization of a Non-normal Code

In this section we consider the construction of a normal code based on a given non-normal network code, that is obtaining a code C satisfying $\det(I_{|E|} - K_C) \neq 0$. We refer to this as *normalization of a non-normal code*.

Let $K_C = (k_{d,e})$ be a network code with associated matrix $M := I_{|E|} - K_C$ where in M the rows and columns represent the edges of network G according to a given order on the edges. The cofactor of entry $m_{i,i}$ is the matrix $M_{i,i}$ obtained from M by the deletion of i th row and i th column of M , that is removing the edge indexed by i

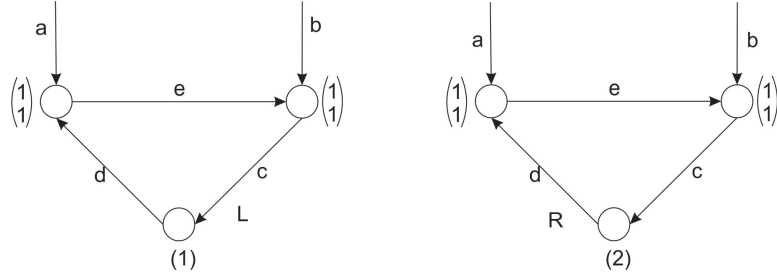


Figure 3.1: A non-normal code (1) and a normal code derived from it (2).

from the network.

Theorem 9. Suppose $K_C = (k_{d,e})$ is a network code on a nonbinary field F_q with associated matrix $M := I_{|E|} - K_C$ satisfying $\det(M) = 0$ over F_q , that is C is not a normal code. Assume that for a nonsource outgoing edge e , that is $e \notin \text{Out}(s)$, indexed by i we have $t_0 := \det(M_{i,i}) \neq 0$. Let C' be the code obtained from C by multiplying all $k_{d,e}$ by a fixed $\alpha \notin \{0, 1\}$. Then C' is a normal code.

Proof. Let $M' := I_{|E|} - K_{C'}$ be the matrix associated with C' . The difference between M and M' is just in their i th column. In fact the i th column of M' , except $m'_{i,i}$ for which we have $m'_{i,i} = m_{i,i} = 1$, is obtained by multiplying the i th column of M by α . Note that we have $k_{d,e} \neq 0$ for at least one edge d since otherwise e has no role in the network and can be removed. If we compute both $\det(M)$ and $\det(M')$ by cofactor expansion along their i th column, it is easy to see that for some t we have $0 = \det(M) = t + \det(M_{i,i}) = t + t_0$ and $\det(M') = \alpha t + \det(M_{i,i}) = \alpha t + t_0$. If $\det(M') = \alpha t + t_0 = 0$ then we get $(1 - \alpha)t = 0$ and hence $t = 0$ since $\alpha \neq 1$; but it follows from $t = 0$ and $0 = \det(M) = t + t_0$ that $t_0 = 0$, a contradiction. Therefore $\det(M') \neq 0$ and C' is normal. \square

Example 15. Consider the left network shown in Fig. 3.1 over F_3 , with $\text{Out}(s) = \{a, b\}$ and $f_a = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $f_b = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. This code does not introduce a set of coding vectors, since if we assume that such coding vectors exist then

$$f_e = f_a + f_d = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + f_c = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix} + f_e \right\} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + f_e,$$

which results in $\begin{pmatrix} 1 \\ 1 \end{pmatrix} = 0$, a contradiction. Note also that for this code C we have the following associated matrix M whose rows (left-to-right) and columns (top-to-down)

correspond to the edges a , b , c , d and e , respectively.

$$M := I_{|E|} - K_C = \begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 2 & 0 & 1 \end{pmatrix}$$

Over F_3 we have $\det(M) = 0$, that is the discriminant of C is zero. Assuming that M_{ij} is the matrix obtained from M by deletion of its i th row and j th column, we see that matrices $M_{3,3}$, $M_{4,4}$ and $M_{5,5}$ are with determinant 1. Now for instance consider the node connecting c and d in the network, that is considering the fourth column of M , and multiply the coding coefficient $k_{c,d} = 1$ by $\alpha = 2$; this gives a code C' shown by Fig. 3.1R. This network has associated matrix M' with $\det(M') = 2$ where

$$M' := I_{|E|} - K_{C'} = \begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 2 & 0 & 1 \end{pmatrix}$$

The code C' gives a unique set of coding vectors as follows. It follows from

$$f_e = f_a + f_d = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 2f_c = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 2\left\{\begin{pmatrix} 0 \\ 1 \end{pmatrix} + f_e\right\} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} + 2f_e,$$

that we must have $f_e = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$, $f_c = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$ and $f_d = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

3.2 Construction of a Linear Multicast Code on a Cyclic Network

In the previous chapter we studied the construction of a linear multicast, broadcast or dispersion code for a cyclic network based on an associated acyclic four-layer network. We showed that a linear multicast, broadcast or dispersion code for the corresponding four-layer acyclic network results in a linear multicast, broadcast or dispersion code for the original cyclic network. The construction process of the corresponding acyclic network is of time-complexity $O(E^2)$ where E is the number of edges in the cyclic network. Algorithm 6 is a direct approach to create a linear multicast, broadcast or dispersion code on a cyclic network which is conjectured to have a running time

of $O(E)$ which is a much better running time and hence this algorithm is better compared to the four layer network algorithm. Algorithm 6 is given below, but prior to introducing it a few definitions and notation are given.

Let G be an ω -dimensional cyclic network with edge set E and the set of multicast nodes T . Though the ω edges in $Out(s)$ are connected at s , but for the purpose of the following algorithm, we ignore this connectivity at s , hence graphically, we consider ω data generating edges each of which has a distinct copy of s as its tail node.

Let $\{p_{t,1}, \dots, p_{t,\omega}\}$ be a set of edge disjoint paths from the source node s to the sink node $t \in T$. Denote by E_t the set of edges in these paths and set $E'_T = \bigcup_{t \in T} E_t$. Any edge e not being in E'_T can be ignored and deleted from G ; hence we assume that $E = E'_T$. Remove any edge $e \in E'_T$ from E'_T if it is only in one edge set E_t , and denote the remaining edge set by E_T and consider the graph $G' := G - E'_T \setminus E_T$, that is G' is obtained from G by deleting any edge e that is only in one edge set E_t . The graph G' may or may not be connected; each connected component of G' will be referred to as a graph-component of G' .

If a component of G' is a path P then any data symbol on the first edge of P can be transferred without any change to other edges of P and the edges in $E'_T \setminus E_T$ that are connected in G to this path, that is the corresponding coding coefficients $k_{d,e}$ can be defined $k_{d,e} = 1$. Hence, P can be dealt with as a length-one path, that is a single edge. In the left graph in Figure 3.2, the edge q is a component and its incoming edges b and c are in $E'_T \setminus E_T$.

Suppose a component G'_1 of G' contains a cycle and has an edge e with in-degree zero in G' ; then it is easily verified that e can be deleted and any element of $Out(e)$ in G can have $tail(e)$ as its tail. It is also easy to see that if G'_1 has a node v with in-degree zero in G'_1 (see $tail(a)$ in the second graph of Figure 3.2), then the edges in $E'_T \setminus E_T$ entering this node v can be partitioned to distinct subsets each assigned to one edge in $Out(v)$; that is a figure like the second graph of Figure 3.2 can be changed to the third graph of this figure. Hence, in this Example, G'_1 can be reduced to a length-three cycle and two edges a and b with $tail(a) \neq tail(b)$. This motivates the following definition.

A directed graph is called closed if every node of the graph has positive in-degree and out-degree. A component G'_1 of G' is called *closed component* if it is a closed graph; the fourth graph in Figure 3.2 is a closed graph. This implies that a component G'_1 of G' is either a single edge, a close component, or it is complex meaning that it consists of a closed graph and a positive number of edges with in-degree zero in

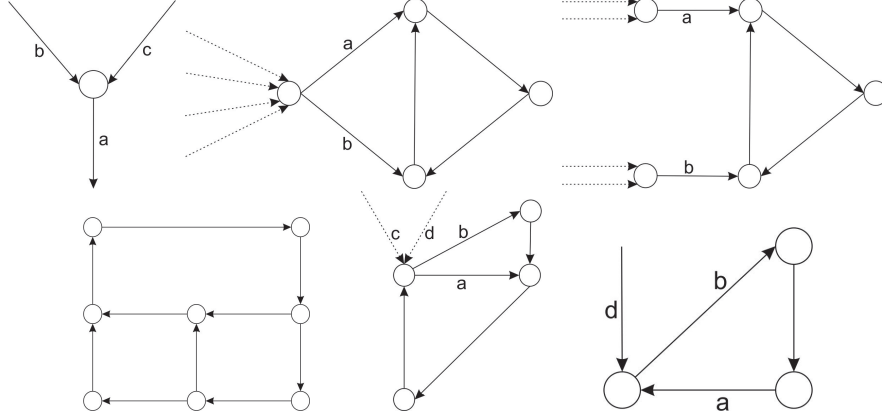


Figure 3.2: Different components of a graph $G' = G - E_T \setminus E_T$.

G' . The last two graphs in Figure 3.2 are of the third type. For simplicity we refer to these as type-1, type-2 and type-3 components, respectively. We refer to a type-3 component G'_1 having α edges with in-degree zero as an $\alpha + 1$ dimensional component and a set of coding vectors assigned to the edges of G'_1 is called acceptable if: all of the edges in the closed part of G'_1 have the same coding vector; any $k \leq \omega$ coding vectors assigned to the edges of G'_1 are linearly independent.

Algorithm 6. Let G be a given ω -dimensional cyclic network.

- Determine the set of multicast nodes T in the cyclic network and call them sink nodes.
- For each sink node $t \in T$ in the network find ω edge disjoint paths $\{p_{t,1}, \dots, p_{t,\omega}\}$ from the source node s to the sink node t . This step has a running time of $O(|T|E)$ which $|T|$ is the number of sink nodes in the network. Denote by E_t the set of edges in these paths and set $E'_T = \bigcup_{t \in T} E_t$.
- Simplify the network by ignoring any edge e that is not in $E'_T = \bigcup_t E_t$.
- For each edge e in E'_T find the number of sink nodes whose associated edge set E_t contain this edge e .
- Remove any edge $e \in E'_T$ from E'_T if it is only in one edge set E_t , and denote the remaining edge set by E_T and store it in a list.
- Find the size of the list and put it in the variable L_1 and refer to the list as $L_{network}$.

- Sort the edges in the list according to their distance from the source node in an increasing order. This step has a running time of $O(E \log E)$ however in this step the number of edges in E_T is much smaller than the number of edges in the original network. Therefore we can consider the running time of this step to be insignificant.
- For each edge in the list, following the given ordering, choose an arbitrary coding vector on the lowest field size such that:
all the edges in a type-2 component G'_1 have the same coding vector,
the coding vectors assigned to any type-3 component is acceptable,
and in any stage of the process, among the assigned coding vectors, any combination of ω coding vectors are linearly independent.
- Intuitively, the coding kernel for a pair (e, e') of adjacent edges is set $k_{e,e'} = 1$ if e' has no incoming edge other than e . Also we set $k_{e,e'} = 1$ if $e \in \text{Out}(s)$.
- If an edge $e \in E'_T \setminus E_T$, or in general a path p , connects two components of G' , and this edge e , or path p , belongs to $p_{t,j}$ for some $t \in T$ and $1 \leq j \leq \omega$, and the edge of $p_{t,j}$ in E_T which precedes e , or p , is d , then we assign the coding vector of d to e (or all edges of the path p).
Consider a sink node t and its ω incoming edges $e_1, e_2, \dots, e_\omega$ that are in E_t , and suppose e_i is on the path $p_{t,i}$. As $\text{Out}(s) \subset E_T$, let e'_i be the last edge of the path $p_{t,i}$ that is in the set E_T , and assume that $p'_{t,i} := e'_i \cdots e_i$, with possibly $e'_i = e_i$, is the sub-path of $p_{t,i}$ which begins with e'_i and ends up at e_i . We assign a routing role to the path $p'_{t,i} = e'_i \cdots e_i$ by defining $k_{e,e'} = 1$ for any adjacent edges on this path. We also set $k_{e,e'} = 0$ if $e, e' \in E'_T \setminus E_T$ and for no $t \in T$ and for no $1 \leq i \leq \omega$, these two edges are adjacent edges on the path $p'_{t,i}$.
- The assigned global coding vectors together with the assigned coding coefficients are used to find the unknown coding coefficients.

As based on the fourth last step of the algorithm the global coding vectors assigned to any ω edges in E_T are linearly independent, it follows from the second last item of the algorithm that $\dim(V_t) = \omega$ for any sink node $t \in T$, and hence the output of this algorithm is a multicast code.

Our proposed algorithm which is given in Algorithm 6 does not necessarily give the best answer in terms of the field size. This algorithm is illustrated by Example 13.

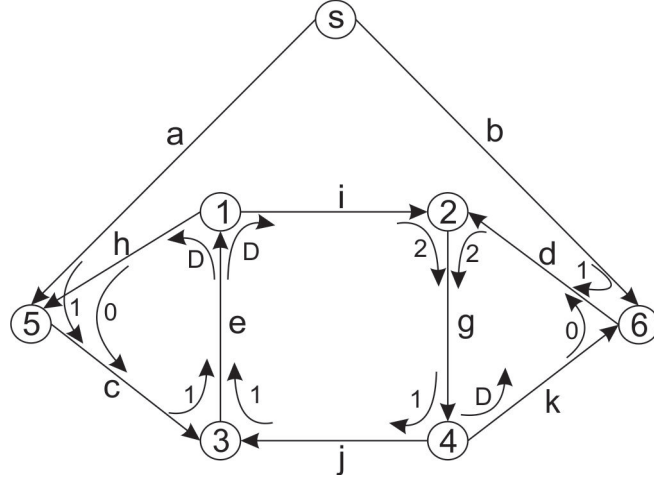


Figure 3.3: Shuttle network with the first set of coding coefficients given in Table 3.2 obtained using Algorithm 6.

Example 16. Consider the shuttle network given in Figure 3.3. It is easy to see that $E'_T = \{a, c, e, i, g, k, b, d, j, h\}$ and $E_T = \{a, b, e, g\}$ and $L_1 = 4$. The coding vectors for E_T are defined below. The defined coding vectors have the required linear independency for a linear multicast code.

We first set $f_a = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $f_b = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Then we set $f_g = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $f_e = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$. Any two vectors of these four vectors are linearly independent over F_3 . As the edge labeled i is in $E'_T \setminus E_T$ and connects two components of E_T and is an edge of one of the two paths from the source s to the sink labeled by 6, we assign to i the global coding vector of its preceding edge on this path in E_T , that is we set $f_i = f_e$. By the same reasoning we set $f_j = f_g$. Using the second paragraph of the second last step of the algorithm, we set $f_h = f_e$ and $f_k = f_g$, that is $k_{e,h} = k_{g,k} = 1$. By the third last command of the algorithm we set $k_{a,c} = k_{b,d} = 1$. Thus the unknown coding coefficients are $k_{h,c}$, $k_{c,e}$, $k_{j,e}$, $k_{k,d}$, $k_{d,g}$, and $k_{i,g}$. As shown below, using the assigned coding vectors and coding coefficients, we determine the coding coefficients of a multicast code. The algorithm generates four multicast codes.

For simplicity let set $a_1 = k_{i,g}$, $a_2 = k_{d,g}$, $a_3 = k_{k,d}$, $a_4 = k_{j,e}$, $a_5 = k_{c,e}$, and $a_6 = k_{h,c}$.

From the structure of the shuttle network we have

$$\begin{cases} f_g = \begin{pmatrix} 1 \\ 1 \end{pmatrix} = a_1 f_i + a_2 f_d = a_1 \begin{pmatrix} 2 \\ 1 \end{pmatrix} + a_2 f_d = a_1 \begin{pmatrix} 2 \\ 1 \end{pmatrix} + a_2 \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix} + a_3 \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\} \\ f_e = \begin{pmatrix} 2 \\ 1 \end{pmatrix} = a_4 f_j + a_5 f_c = a_4 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + a_5 \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix} + a_6 f_e \right\} = a_4 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + a_5 \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix} + a_6 \begin{pmatrix} 2 \\ 1 \end{pmatrix} \right\}. \end{cases}$$

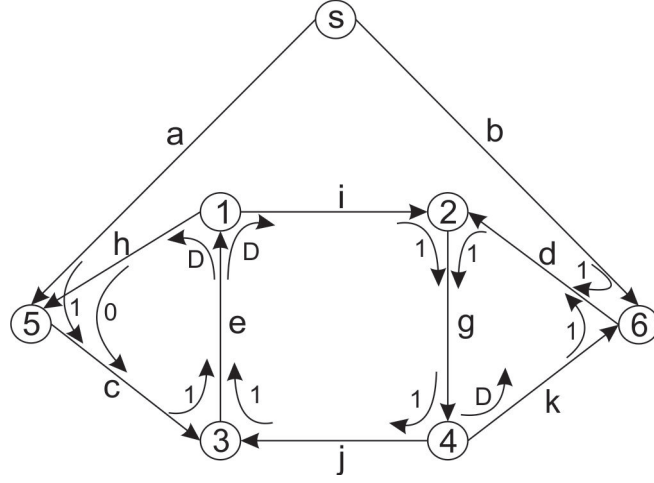


Figure 3.4: Shuttle network with the coding coefficients obtained by the four layer network algorithm.

This is simplified to

$$\begin{cases} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = a_1 \begin{pmatrix} 2 \\ 1 \end{pmatrix} + a_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + a_3 a_4 \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ \begin{pmatrix} 2 \\ 1 \end{pmatrix} = a_5 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + a_6 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + a_5 a_6 \begin{pmatrix} 2 \\ 1 \end{pmatrix} \end{cases}$$

which gives

$$\begin{cases} (1 - a_3 a_4) \begin{pmatrix} 1 \\ 1 \end{pmatrix} = a_1 \begin{pmatrix} 2 \\ 1 \end{pmatrix} + a_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ (1 - a_6 a_5) \begin{pmatrix} 2 \\ 1 \end{pmatrix} = a_5 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + a_6 \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \end{cases}$$

Each of these equations has two sets of solutions given by

$$\begin{cases} a_1 = a_2 = 2, a_3 = 0, & \text{and} & a_1 = a_2 = 1, a_3 = 2 \\ a_4 = a_5 = 1, a_6 = 0, & \text{and} & a_4 = a_5 = 2, a_6 = 1. \end{cases}$$

It is easy to check that any solution of the first equation together with any solution of the second equation gives a multicast code for which we have $f_a = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $f_b = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $f_g = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, and $f_e = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$. These four codes are given in Table 3.2. Note that as the network is cyclic, to have a casual code we need to have at least one delay on each cycle; hence we have multiplied the coding coefficients $k_{e,i}$, $k_{e,h}$ and $k_{g,k}$ by D . The global coding vectors derived by the code given in the first row of Table 3.2 is given in Table 3.4.

If we use Algorithm 3 and the algorithm proposed in [4], the coding coefficients given in Table 3.3 are obtained for the shuttle network. Therefore, our proposed algorithm gives codes on F_3 , but the optimal answer is on F_2 . Although the proposed

Table 3.1: Algorithm 6 and the four layer network algorithm executed on the networks in Figures 3.5 to 3.9.

Algorithm Network	Four layer network algorithm	Algorithm 6
Figure 3.5	F_2	F_2
Figure 3.6	F_2	F_2
Shuttle network	F_2	F_3
Figure 3.7	F_2	F_5
Figure 3.8	F_2	F_2
Figure 3.9	F_2	F_3

algorithm may not give the optimal answer in terms of the field size, it will give us a reasonable solution to the problem. The coding vectors for each case is given in Tables 3.4 and 3.5.

Consider the networks in Figures 3.5 to 3.7. The four layer network algorithm and Algorithm 6 have been executed on each of the networks and the results are given in Table 3.1. Table 3.1 shows that Algorithm 6 gives a solution in a field such that the field size is close to optimal.

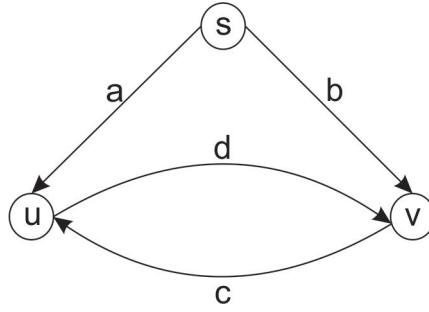


Figure 3.5: The first cyclic network for Table 3.1.

Example 17. Consider the network shown in Figure 3.9; the corresponding graph has single-edge components a , b , and c , and the cycle component C consisting of edges labeled 10, 11, and 12. Hence we assign coding vectors $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$, and $(1, 1, 1)$, over F_3 to components a , b , c , and C , respectively. Any $\omega = 3$ element subset of these coding vectors are linearly independent. We set $k_{a,7} = k_{b,8} = k_{c,11} = k_{10,11} = k_{c,9} = k_{9,12} = k_{11,12} = 1$; this together with $f_{10} = (1, 1, 1)$ gives $f_{11} = (1, 2, 1)$, $f_{12} = (1, 2, 2)$, and $k_{7,10} = k_{12,10} = 2$. Then we set $f_{13} = f_{12}$, $f_{14} = f_{10}$, and

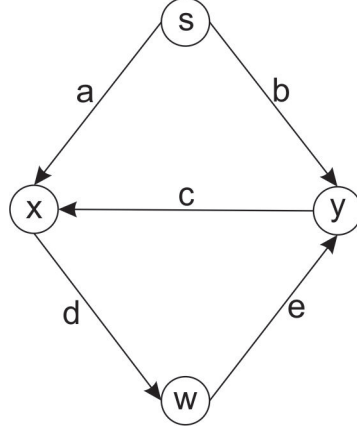


Figure 3.6: The second cyclic network for Table 3.1.

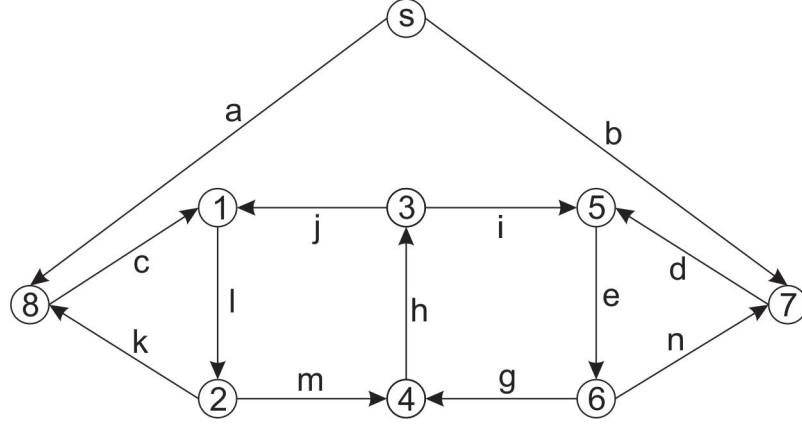


Figure 3.7: The fourth cyclic network for Table 3.1.

$f_{15} = f_{11}$. By giving a delay D to the cycle component C at $(12, 10)$, that is setting $k_{12,10} = 2D$, we get $f_{10} = (2, 0, 0) + 2Df_{12}$, $f_{11} = (0, 1, 0) + f_{10} = (2, 1, 0) + 2Df_{12}$, $f_{12} = (0, 0, 1) + f_{11} = (2, 1, 1) + 2Df_{12}$, which gives

$$f_{12} = \frac{1}{1+D}(2, 1, 1), \quad f_{10} = \frac{2}{1+D}(1, D, D), \quad f_{11} = \frac{1}{1+D}(2, 1, 2D).$$

3.3 An Improvement of the Four Layer Network Algorithm

In the previous section we introduced a new algorithm for generating a linear multicast code for a cyclic network. In this section we give a theorem which improves the four layer network algorithm.

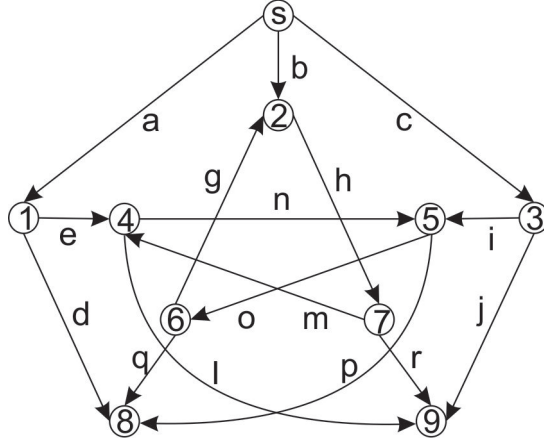


Figure 3.8: The fifth cyclic network for Table 3.1.

Table 3.2: Four sets of coding coefficients for the shuttle network obtained by Algorithm 6.

$k_{a,c}$	$k_{h,c}$	$k_{c,e}$	$k_{j,e}$	$k_{i,g}$	$k_{d,g}$	$k_{b,d}$	$k_{k,d}$	$k_{e,i}$	$k_{e,h}$	$k_{g,j}$	$k_{g,k}$
1	0	1	1	2	2	1	0	D	D	1	D
1	0	1	1	1	1	1	2	D	D	1	D
1	1	2	2	2	2	1	0	D	D	1	D
1	1	2	2	1	1	1	2	D	D	1	D

Theorem 10. Let G be a cyclic network with at least two sink nodes. Then in the four layer network algorithm, the node s_4 and its incoming edges in the acyclic network G' can be removed.

Proof. Let C' be a multicast code on the acyclic network G' ; the node s_4 in G' is a sink node and the reception of full data by s_4 guarantees the non-singularity of the code C on G derived from C' (Theorem 19 in [3]). This condition of full data reception by s_4 can also be satisfied if for each $e \in E \setminus \text{Out}(s)$ there is one edge e' from $e_{(3)}$ to a sink v_4 , distinct from s_4 , in G' , and also the coding vector of these edges are linearly independent.

Consider an edge $e \in E \setminus \text{Out}(s)$ in the cyclic network G . This edge is the input to at most one of the sink nodes. Therefore in the acyclic network G' there are two possible cases. The first case is that $e_{(3)}$ is connected to all sinks in the acyclic network and the second case is that it is connected to all sinks except one of them in the acyclic network. Therefore, $\deg(\text{out}(e_{(3)})) = d - 1$ or d , where d is the number of sinks in the cyclic network G . If the number of sinks in G is at least two, then $\deg(\text{out}(e_{(3)})) \geq 1$.

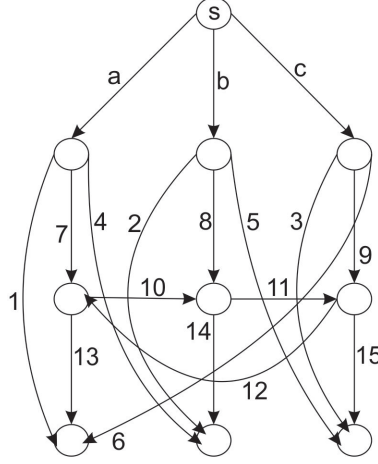


Figure 3.9: A cyclic network G which has three single-edge components and one cycle component.

Table 3.3: Coding coefficients derived from the four layer network algorithm for the shuttle network.

$k_{a,c}$	$k_{h,c}$	$k_{c,e}$	$k_{j,e}$	$k_{i,g}$	$k_{d,g}$	$k_{b,d}$	$k_{k,d}$	$k_{e,i}$	$k_{e,h}$	$k_{g,j}$	$k_{g,k}$
1	0	1	1	1	1	1	1	D	D	1	D

In this case, for each $e \in E \setminus \text{Out}(s)$ we may consider one outgoing edge e' from $e_{(3)}$ to a sink v_4 , distinct from s_4 , and then ask for the linear independency of the coding vector of these chosen edges. \square

The benefit of this theorem is that reduction of sink nodes in a network simplifies the network and coding on the reduced network in general will be easier than the original network.

Example 18. Consider the cyclic network G and its corresponding acyclic four layer network G' given in Figure 3.10. The removal of node s_4 in G' results in the third graph given in Figure 3.10. In this acyclic network G' we have $\deg(c_{(3)}) = 1$, $\deg(d_{(3)}) = 2$ and $\deg(e_{(3)}) = 1$. Hence we have to consider either the edge set labeled by 18, 19, 22, or the edge set labeled by 18, 21, 22. In any multicast code for this network the coding vectors f_{18} and f_{19} (also the coding vectors f_{21} and f_{22}) are linearly independent. Based on the given theorem, in order to get a multicast code for G , we need to have a multicast code C' on this acyclic network such that in C' at least the coding vectors $\{f_{18}, f_{19}, f_{22}\}$ or $\{f_{18}, f_{21}, f_{22}\}$ are linearly independent.

Table 3.4: Coding vectors of the first code given in Table 3.2.

f_a	f_b	f_c	f_d	f_e	f_g	f_h	f_i	f_j	f_k
$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 2 \\ 1+D \end{pmatrix}$	$\begin{pmatrix} \frac{2D}{1+D} \\ \frac{D}{2} \\ \frac{D}{1+D} \end{pmatrix}$	$\begin{pmatrix} \frac{D}{1+D} \\ \frac{D}{2+2D} \end{pmatrix}$	$\begin{pmatrix} \frac{D}{1+D} \\ \frac{D}{2+2D} \end{pmatrix}$	$\begin{pmatrix} \frac{2D}{1+D} \\ \frac{D}{2} \\ \frac{D}{1+D} \end{pmatrix}$	$\begin{pmatrix} \frac{2D^2}{1+D} \\ \frac{2D}{1+D} \end{pmatrix}$

Table 3.5: Coding vectors of the code given in Table 3.3 for the shuttle network.

f_a	f_b	f_c	f_d	f_e	f_g	f_h	f_i	f_j	f_k
$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} D^2 \\ D \end{pmatrix}$	$\begin{pmatrix} 1+D \\ 1+D \\ D \end{pmatrix}$	$\begin{pmatrix} D \\ 1+D \\ D \end{pmatrix}$	$\begin{pmatrix} D^2+D \\ 1+D \end{pmatrix}$	$\begin{pmatrix} D^2+D \\ 1+D \end{pmatrix}$	$\begin{pmatrix} D \\ 1+D \\ D \end{pmatrix}$	$\begin{pmatrix} D^2 \\ 1+D \end{pmatrix}$

The network matroid algorithm was introduced in Section 2.1. As mentioned, on the four layer acyclic network G' the coding vectors $\{f_{17}, f_{18}, f_{19}\}$ must be linearly independent. Based on the network matroid algorithm columns 17, 18 and 19 of the global coding vector matrix F , consisting of all the global coding vectors, must give a full rank matrix. This property must be satisfied by columns 20, 21 and 22 of F , and also by the last three columns of F . Columns 17 to 25 of the F matrix are shown below as $F = (AB)$ where

$$A = \begin{pmatrix} 0 & k_{1,5}k_{5,14}k_{14,18} & 0 & k_{1,4}k_{4,13}k_{13,20} & 0 \\ k_{2,7}k_{7,12}k_{12,17} & k_{2,8}k_{8,14}k_{14,18} & k_{2,9}k_{9,15}k_{15,19} & 0 & k_{2,9}k_{9,15}k_{15,21} \\ 0 & 0 & k_{3,10}k_{10,15}k_{15,19} & 0 & k_{3,10}k_{10,15}k_{15,21} \end{pmatrix} \text{ and}$$

$$B = \begin{pmatrix} k_{1,6}k_{6,16}k_{16,22} & k_{1,5}k_{5,14}k_{14,23} & 0 & k_{1,6}k_{6,16}k_{16,25} \\ 0 & k_{8,14}k_{14,23} & k_{2,9}k_{9,15}k_{15,24} & 0 \\ k_{3,11}k_{11,16}k_{16,22} & 0 & k_{3,10}k_{10,15}k_{15,24} & k_{3,11}k_{11,16}k_{16,25} \end{pmatrix}$$

Hence the related three matrices H_i , which are needed to be full rank, are as follows.

$$H_1 = \begin{pmatrix} 0 & k_{1,5}k_{5,14}k_{14,18} & 0 \\ k_{2,7}k_{7,12}k_{12,17} & k_{2,8}k_{8,14}k_{14,18} & k_{2,9}k_{9,15}k_{15,19} \\ 0 & 0 & k_{3,10}k_{10,15}k_{15,19} \end{pmatrix}$$

$$H_2 = \begin{pmatrix} k_{1,4}k_{4,13}k_{13,20} & 0 & k_{1,6}k_{6,16}k_{16,22} \\ 0 & k_{2,9}k_{9,15}k_{15,21} & 0 \\ 0 & k_{3,10}k_{10,15}k_{15,21} & k_{3,11}k_{11,16}k_{16,22} \end{pmatrix}$$

$$H_3 = \begin{pmatrix} k_{1,5}k_{5,14}k_{14,23} & 0 & k_{1,6}k_{6,16}k_{16,25} \\ k_{8,14}k_{14,23} & k_{2,9}k_{9,15}k_{15,24} & 0 \\ 0 & k_{3,10}k_{10,15}k_{15,24} & k_{3,11}k_{11,16}k_{16,25} \end{pmatrix}$$

Therefore, we need to have

$$\begin{aligned} & k_{1,5}k_{5,14}k_{14,18}k_{2,7}k_{7,12}k_{12,17}k_{3,10}k_{10,15}k_{15,19} \neq 0, \\ & k_{1,4}k_{4,13}k_{13,20}k_{2,9}k_{9,15}k_{15,21}k_{3,11}k_{11,16}k_{16,22} \neq 0, \\ & k_{1,5}k_{5,14}k_{14,23}(k_{2,9}k_{9,15}k_{15,24}k_{3,11}k_{11,16}k_{16,25}) \\ & + k_{1,6}k_{6,16}k_{16,25}(k_{2,8}k_{8,14}k_{14,23}k_{3,10}k_{10,15}k_{15,24}) \neq 0. \end{aligned}$$

The related three H_i matrices for the network $G' - \{s_4\}$ which need to be full rank are:

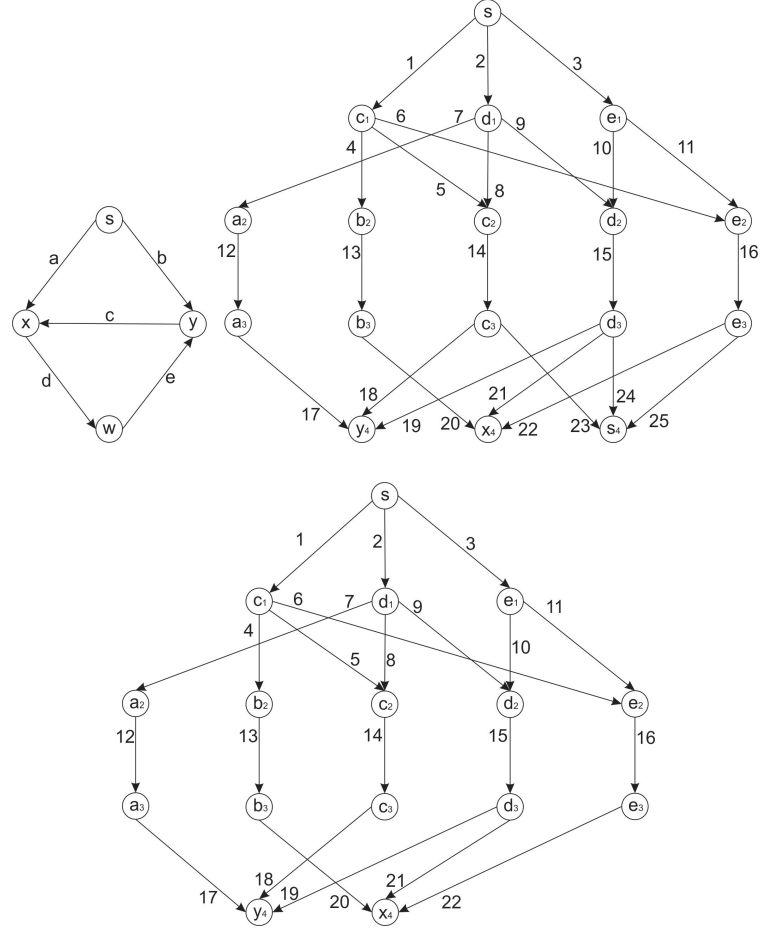


Figure 3.10: A cyclic network G and its associated four layer network G' . The third graph is the acyclic network $G' - \{s_4\}$.

$$\begin{aligned}
 H'_1 &= \begin{pmatrix} 0 & k_{1,5}k_{5,14}k_{14,18} & 0 \\ k_{2,7}k_{7,12}k_{12,17} & k_{2,8}k_{8,14}k_{14,18} & k_{2,9}k_{9,15}k_{15,19} \\ 0 & 0 & k_{3,10}k_{10,15}k_{15,19} \end{pmatrix} \\
 H'_2 &= \begin{pmatrix} k_{1,4}k_{4,13}k_{13,20} & 0 & k_{1,6}k_{6,16}k_{16,22} \\ 0 & k_{2,9}k_{9,15}k_{15,21} & 0 \\ 0 & k_{3,10}k_{10,15}k_{15,21} & k_{3,11}k_{11,16}k_{16,22} \end{pmatrix} \\
 H'_3 &= \begin{pmatrix} k_{1,5}k_{5,14}k_{14,18} & 0 & k_{1,6}k_{6,16}k_{16,22} \\ k_{2,8}k_{8,14}k_{14,18} & k_{2,9}k_{9,15}k_{15,19} & 0 \\ 0 & k_{3,10}k_{10,15}k_{15,19} & k_{3,11}k_{11,16}k_{16,22} \end{pmatrix}
 \end{aligned}$$

These matrices need to have nonzero determinant, hence we need to have

$$\begin{aligned}
 &k_{1,5}k_{5,14}k_{14,18}k_{2,7}k_{7,12}k_{12,17}k_{3,10}k_{10,15}k_{15,19} \neq 0, \\
 &k_{1,4}k_{4,13}k_{13,20}k_{2,9}k_{9,15}k_{15,21}k_{3,11}k_{11,16}k_{16,22} \neq 0, \\
 &k_{1,5}k_{5,14}k_{14,18}(k_{2,9}k_{9,15}k_{15,19}k_{3,11}k_{11,16}k_{16,22}) \\
 &+ k_{1,6}k_{6,16}k_{16,22}(k_{2,8}k_{8,14}k_{14,18}k_{3,10}k_{10,15}k_{15,19}) \neq 0.
 \end{aligned}$$

3.4 Creating a Generic Network Code on a Cyclic Network

In this section a new approach is taken to create a generic network code for a cyclic network G without employing the de-cycling process, that is without making use of the related four layer acyclic network. The authors in [7] defined a normal F -linear network code and developed a theorem which is given below.

Theorem 11. [7] A normal F -linear network code $K = (k_{d,e})$ on a cyclic network G with edge set E is generic if and only if every path independent set of edges B in E is linearly independent with respect to $k_{d,e}$, that is the global coding vectors of the elements of B are linearly independent.

Based on this theorem one can construct a generic network code for a given cyclic network G by assigning linearly independent global coding vectors to path independent edges of G . For this, given a network G , first we assign linearly independent coding vectors to path independent edges, then the assigned coding vectors are used to find a set of coding coefficients, and finally using the obtained coding coefficients and the delays assigned to the cycles of the network we determine a generic code for the given network.

Example 19. Consider the shuttle network given in Figure 3.3 but without its coding coefficients. We set $f_a = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $f_b = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Consider the path $\{a, c\}$ and edge b and consider the edge a and the path $\{b, d, g, j, e, h, c\}$. This shows that the edge c is path independent from both a and b . Therefore, the coding vector f_c is set to $f_c = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Consider edge disjoint paths $\{a, c, e, i, g, k, d\}$ and edge b , edge disjoint paths $\{b, d\}$ and a , and edge disjoint paths $\{a, c\}$ and $\{b, d\}$. It follows from the existence of these paths that the edge d is path independent from a, b and c . Therefore, the coding vector of f_d must be linearly independent of those assigned to a, b and c ; hence we consider coding over F_3 and set $f_d = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$. Using the same reasoning, we consider coding over F_5 and set $f_e = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$ and $f_g = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$. The edges h and i have only e as their incoming edge, and hence we set $f_h = f_i = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$. Similarly, we define $f_j = f_k = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$.

Based on these coding vectors, the coding coefficients given in Table 3.6 are obtained for the shuttle network. To each cycle a delay coefficient D must be inserted and the coding vectors must be recomputed with the delays in each cycle. Table 3.7 shows these coding coefficients and the position of the assigned delay to each cycle.

Table 3.6: Coding coefficients derived from coding vectors given in Example 19, assigned to the network shown in Figure 3.3.

$K_{a,c}$	$K_{h,c}$	$K_{c,e}$	$K_{j,e}$	$K_{e,h}$	$K_{e,i}$	$K_{i,g}$	$K_{d,g}$	$K_{g,j}$	$K_{g,k}$	$K_{k,d}$	$K_{b,d}$
3	1	2	2	1	1	4	3	1	1	4	3

Table 3.7: Coding coefficients related to Table 3.6 after inserting a delay in each cycle of the shuttle network. The location of D shows the position of the assigned delays.

$K_{a,c}$	$K_{h,c}$	$K_{c,e}$	$K_{j,e}$	$K_{e,h}$	$K_{e,i}$	$K_{i,g}$	$K_{d,g}$	$K_{g,j}$	$K_{g,k}$	$K_{k,d}$	$K_{b,d}$
3	1	2	2	D	1	4	3	D	D	4	3

It follows from

$$\begin{cases} f_e = 2f_c + 2f_j = 2\{3\binom{1}{0} + Df_e\} + 2Df_g = \binom{1}{0} + 2Df_e + 2Df_g \\ f_g = 3f_d + 4f_i = 3\{3\binom{0}{1} + 4Df_g\} + 4f_e = \binom{0}{4} + 2Df_g + 4f_e \end{cases}$$

that

$$\begin{cases} 2Df_g = (1 + 3D)f_e + \binom{4}{0} \\ (1 + 3D)f_g = \binom{0}{4} + 4f_e. \end{cases}$$

This gives $f_i = f_e = \frac{1}{4D^2+3D+1}\binom{1+3D}{3D}$, $f_g = \frac{1+3D}{2D}f_e + \frac{1}{2D}\binom{4}{0} = \frac{1}{4D^2+3D+1}\binom{4}{4+2D}$. Hence $f_j = f_k = Df_g = \frac{1}{4D^2+3D+1}\binom{4D}{4D+2D^2}$; $f_d = \binom{0}{3} + 4f_k = \frac{1}{4D^2+3D+1}\binom{D}{3}$; $f_h = Df_e = \frac{1}{4D^2+3D+1}\binom{D+3D^2}{3D^2}$, and $f_c = 3f_a + f_h = \frac{1}{4D^2+3D+1}\binom{3}{3D^2}$.

Chapter 4

Conclusions

In this thesis network coding was introduced such that an intermediate node v receives a message from its incoming channels and can transmit a combination of the received messages. The importance of network coding is that it can improve the throughput of the network.

Networks can be divided into cyclic and acyclic networks. For each network a different approach must be taken to create network codes. A linear multicast, broadcast and dispersion code was defined in chapter one. The Jaggi-Sanders algorithm and the network matroid algorithm was given in chapter two to create a linear multicast code on an acyclic network.

For a cyclic network, two different methods can be used to create a linear multicast code. The cyclic network can be decycled and a linear multicast code can be created on the acyclic network which will give a linear multicast code on the cyclic network. Conversely, the cyclic network can be used directly to create the required network code.

The four layer network algorithm in the second chapter was used to associate an acyclic network with the cyclic network. This algorithm has a running time of $O(E^2)$ which is not good for a big network. A new algorithm in Chapter 3 was given to create a linear multicast code with a better running time which is conjectured to be $O(E)$ that gives a solution in a field which the field size is close to optimal. In Chapter 3 an algorithm is provided which uses the cyclic network directly to create a generic network code.

Once the network code has been constructed for a specific network, the network code may not be normal. Therefore, the network code must be normalized. In Chapter 3 a theorem is provided to normalize a non-normal code.

In this thesis our focus was on networks which consists of error-free point to point channels. In a typical network all point to point channels are not necessarily error-free. For future work, the algorithm for creating a linear multicast code and the algorithm for creating a generic network code in Chapter 3 could be extended for networks which have communication channels that are not error-free.

Bibliography

- [1] R. Ahlswede, N. C. 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. 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.
- [3] S. Y. R. Li and Q. T. Sun, “Network Coding Theory Via Commutative Algebra,” *IEEE Transactions on Information Theory*, vol. 57, no. 1, pp. 403–415, 2011.
- [4] 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.
- [5] L. S. Pitsoulis, *Topics in Matroid Theory*. Springer, 2013.
- [6] Q. Sun, S. T. Ho, and S. Y. R. Li, “On Network Matroids and Linear Network Codes,” in *IEEE International Symposium on Information Theory*, 2008, pp. 1833–1837.
- [7] Q. T. Sun, S. Y. R. Li, and C. Chan, “Matroidal Characterization of Optimal Linear Network Codes Over Cyclic Networks,” *IEEE Communications Letters*, vol. 17, no. 10, pp. 1992–1995, 2013.
- [8] C. Yang, C. Ming, and J. Huang, “On Linear Network Coding and Matroid,” in *Advances in Electrical Engineering and Automation*, 2012, vol. 139, pp. 313–321.
- [9] R. W. Yeung, *Information Theory and Network Coding*. Springer, 2008.