A Unified Framework for Linear Network Coding

TAN, Min

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Information Engineering

September 2008

Abstract of thesis entitled:
A Unified Framework for Linear Network Coding
Submitted by TAN, Min
for the degree of Master of Philosophy
at The Chinese University of Hong Kong in September 2008

Network coding is a major breakthrough in modern information theory. The revolutionary idea of performing coding rather than store-and-forward at the intermediate nodes greatly improves network information transmission capacity. More surprisingly, this capacity can be achieved by linear network codes constructible in polynomial time. However, the concept of generic network code, which is the strongest format of linear network code, is still not well understood. In this thesis, we aim to give more explicit interpretations of generic network codes. A condition regarding linear independence among global encoding kernels is given. Based on this condition, alternative definitions of generic network codes are proposed and generic network codes are proved to be the best linear network codes in terms of linearly independence; a unified framework of linear network codes is proposed. This unified framework is used to simplify some existing results. The results of this work can be potentially applied to the static network codes and network error-correcting codes.

摘要

網路編碼是現代資訊理論領域的重大突破。它一改過去只在網路中間節點做存儲和轉發的傳統思想，革命性的引進了在網路中間節點進行編碼的概念，極大的提高了網路的傳輸率。更令人吃驚的是，研究工作表明我們可以在多項式時間內構造出達到多播傳輸極限的線性碼。但是我們對通用碼，一種最強形式的線性碼，卻沒有更好的理解。

在本論文中，我們試圖尋找對通用碼的更加明確的解釋。我們找到了全局編碼向量之間互相獨立的一個條件。基於此條件，我們找到了通用碼的其他幾種等價定義並且證明了在線性獨立意義上通用碼是最好的線性碼。我們還構造了一個線性碼的統一框架；文獻中出現的幾種類別的線性碼都能夠統一在此框架之下。最後，我們還簡化了以前文獻中的一些結果。本論文中的結果可能被應用到靜態網路碼和網路改錯碼的研究中。

Acknowledgement

I wish to express my gratitude to my supervisor, Prof.Raymond Yeung, for his guidance during the study of my Master of Philosophy in Information Engineering. I would like to thank Prof.Bob Li and Prof.Sid Jaggi for their comments and discussions.

I would also like to thank my labmates and friends, Shenghao Yang, Salis Fang, Machael Ngai, Qiang Zhu, King-Ho Lee, Qifu Sun, Zizou Wang, Ziyu Shao, Siu-Ting Ho, Jing Nie, and Ceng Xu with whom I spent a wonderful time.

Last but not least, I would like to thank my parents and girlfriend for their support and love.

This work is dedicated to my dear parents.

Contents

Abstract **i**

Acknowledgement **iii**

1 Introduction **1**
1.1 Previous Work 1
1.2 Motivation 2
1.3 Contributions 2
1.4 Thesis Organization 3

2 Linear Network Coding Basics **5**
2.1 Formulation and Example 5
2.2 Some Notations 9

3 A Unified Framework **13**
3.1 Generic Network Codes Revisited 13
3.2 A Unified Framework 24
3.3 Simplified Proofs 29

4 Conclusion **33**

Bibliography **35**

List of Figures

2.1 The butterfly network 7
2.2 Independent set and associated flow 10

3.1 For single independent set, linear independence can be achieved by routing alone. 17
3.2 Routing in general fails to achieve the desired independence for multiple independent sets. 17
3.3 Graph with directed path 25
3.4 Drawbacks of node-based approach 26
3.5 Linear multicast and regular independent set . . . 30
3.6 Illustration of an extended graph 31
3.7 A linear dispersion on G_E implies a generic network code on G 32

Chapter 1

Introduction

Summary

Review of previous work, motivation and contribution of this thesis are given

1.1 Previous Work

Since the establishment of Shannon theory, people have been working hard to find various information transmission capacities for more than sixty years. The area of single-user channel capacity is well understood. The capacity of both discrete memoryless channel and Gaussian channel are known. However, our knowledge of network information theory is still limited. For example, the capacity of the broadcast channel and the two-way channel are still unknown. In the late-1990's, a major breakthrough in network information theory was made by Ahlswede et al. [1]. In this work, the concept of network coding is introduced and the capacity of the single-source multicast network

was found. Following Ahlswede et al. [1], Li et al. [6] proved that this single-source multicast capacity can be achieved by linear network codes. Generic network codes were introduced as capacity-achieving codes in the same paper. Jaggi et al. [4] further proved that capacity achieving linear network codes can be constructed in polynomial time. Ho et al. [3] showed that random linear network codes can achieve multicast capacity with high probability provided the field size is large enough. Yeung et al. [9] defined different classes of linear network codes, namely generic network codes, linear dispersion, linear broadcast, and linear multicast. They also provided a construction algorithm for generic network codes. Kwok el at. [5] discussed the relationship between generic network codes and linear dispersion.

1.2 Motivation

The generic network code in [6] is originally defined using abstract algebra. This makes its definition difficult to understand. Also, the symmetrical structure of this definition makes it difficult to verify whether a linear network code is generic or not. These two points will be explained in details in the later part of this thesis. Thus, we are motivated to further investigate this concept with the aim to make it more transparent. As we will seee, this leads to alternative definitions of generic network codes that are useful in different contexts.

1.3 Contributions

The main contributions of this paper are summarized in the following:

1. Fundamental concepts regarding linear independence among global encoding kernels are studied in depth and a condition that governs the possibility and impossibility of linear independence among global encoding kernels is given.

2. Based on the condition in (1), the relationship between generic network codes and graph theory is established and alternative definitions of generic network codes are presented.

3. A unified framework for linear network codes based on the condition in (1) is presented.

4. Some exiting results whose original proofs were complicated can be greatly simplified by using this unified framework.

1.4 Thesis Organization

This thesis is organized as follows. In Chapter 2, the basic concept of linear network codes is reviewed and some new definitions are introduced for the convenience of discussion. In Chapter 3, generic network codes are revisited; the disadvantages of the original definition of such codes are discussed; new definitions of generic network codes are introduced and their equivalence to the original definition is proved; we also use the insight developed in this chapter to simplify several existing results. The conclusion of this thesis is in chapter 4. Most results presented in this thesis is based on the author's published paper [8].

□ **End of chapter.**

Chapter 2

Linear Network Coding Basics

Summary

Basic concept of linear network codes are reviewed and some necessary notations are introduced

2.1 Formulation and Example

A communication network is modeled as a finite directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of edges connecting these nodes. A edge in $\mathcal{E}$ will also be referred to as a channel. A node is called a source node if it does not contain any incoming edge; a node is called a sink node if it does not contain any outgoing edge. If the communication network does not contain any directed cycle, then it is called an acyclic network. Otherwise, it is called a cyclic network. If the communication network contains only one source node, then it is called a single-source network. If it contains multiple sources, then it is called a multi-source network. The discussion in this paper is

restricted to single-source acyclic networks. The unique source node is denoted by s and the set of all sink nodes is denoted by T. At the source node s, information to be transmitted across the network is generated. To facilitate our discussion, we assume that multiple edges are allowed between nodes and each edge has unit capacity, which means that one symbol taken from a certain finite field $GF(q)$ can be transmitted over each edge. This assumption is general because we can always quantize the capacity to arbitrary degree of accuracy and represent it by multiple edges. We denote by $In(v)$ the set of incoming edges of node v and $Out(v)$ the set of outgoing edges of node v. We denote by $Tail(e) = t$ if edge e is an outgoing edge of node t and by $Head(e) = t$ if edge e is an incoming edge of node t.

Let the information to be transmitted from the source node be represented by a row vector x which consists of ω symbols in $GF(q)$. Following [9], we install a set of ω incoming imaginary edges at s and associate each of them with a distinct vector in an ω-dimensional standard basis. These vectors are refereed to as the global encoding kernels of the imaginary edges.

The set of all local encoding kernels $k_{d,e} \in GF(q)$, where $d \in In(v)$ and $e \in Out(v)$ for some $v \in \mathcal{V}$, specifies a *linear network code*. For each edge e other than an imaginary edge, we iteratively define its global encoding kernel by

$$f_e = \sum_{d \in In(t)} k_{d,e} f_d, \tag{2.1}$$

where $t = Tail(e)$. In other words, at each intermediate node, the incoming global kernels are linearly combined to produce the outgoing global encoding kernels. The received information symbol at each edge e can be calculated as $x \cdot f_e$. The above

Figure 2.1: The butterfly network

concepts are best explained by an example.

Example 1 *Figure 2.1 is the butterfly network. The two vectors at the source are elements of the standard basis that are associated with the imaginary channels. All other vectors are the local encoding kernels. The vector* $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ *is the global encoding kernel of the imaginary channel* e_{s_1} *and the vector* $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ *is the global encoding kernel of the imaginary channel* e_{s_2}*. The local encoding kernel of node 1, 2 and 4 is* $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ *; the local encoding kernel of node 3* $\begin{bmatrix} 1 & 1 \end{bmatrix}$*; and the local encoding kernel of*

the source node is $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. *The global encoding kernels of each edge can be calculated in an upstream-to-downstream manner by using Formula 2.1. For example, the global encoding kernel of* e_1 *and* e_2 *can be calculated as*

$$\begin{bmatrix} f_{e_1} & f_{e_2} \end{bmatrix} = \begin{bmatrix} f_{e_{s1}} & f_{e_{s2}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{2.2}$$

Thus, $f_{e_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ *and* $f_{e_2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. *The global encoding kernels of other edges can be calculated in a similar manner. The results are listed below:*

$$f_{e_3} = f_{e_5} = f_{e_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{2.3}$$

$$f_{e_2} = f_{e_4} = f_{e_6} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.4}$$

$$f_{e_7} = f_{e_8} = f_{e_9} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{2.5}$$

For each edge e, *the received information* y_e *can be calculated as* $y_e = x \cdot f_e$ *where* x *is the source information which is a 2-dimensional row vector. Based on the received information, the sink nodes then can decode the source information if the local encoding kernels are designed properly. In this example, we have*

$$\begin{bmatrix} y_{e_5} & y_{e_8} \end{bmatrix} = x \cdot \begin{bmatrix} f_{e_5} & f_{e_8} \end{bmatrix} \tag{2.6}$$

$$\begin{bmatrix} y_{e_6} & y_{e_9} \end{bmatrix} = x \cdot \begin{bmatrix} f_{e_6} & f_{e_9} \end{bmatrix}. \tag{2.7}$$

We note that $\begin{bmatrix} f_{e_5} & f_{e_8} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ *and* $\begin{bmatrix} f_{e_6} & f_{e_9} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$ *are all invertible matrices. Thus, we can always recover the source information at nodes 5 and 6 by inverting these two matrices respectively.*

2.2 Some Notations

For a collection of nodes T, we define

$$V_T = \langle f_e : Head(e) \in T \rangle.$$

For a set of edges E, we denote their corresponding global encoding kernels by

$$K(E) = \{f(e) : e \in E\}.$$

A sequence of edges $e_1, e_2, \ldots, e_n$, where e_1 may be an imaginary channel, form a *path* if $Head(e_i) = Tail(e_{i+1})$ for $1 \leq i \leq n-1$. Two paths are *edge-disjoint* if they do not have any edge in common.

A set of edges is an *independent set*[1] if each edge is on a path originating from an imaginary channel (i.e., the first edge of the path is an imaginary channel) and these paths are edge-disjoint. We call this set of paths an *associated flow* for this independent set. Note that an independent set concerns only the position of edges in the graph but not the global encoding kernels that may be assigned to them and the global encoding kernels of an independent set can be linearly dependent. Also an independent set may have more than one associated flows.

[1]This name is justified in a separate paper [7] which explicitly defines the underlying matroid structure.

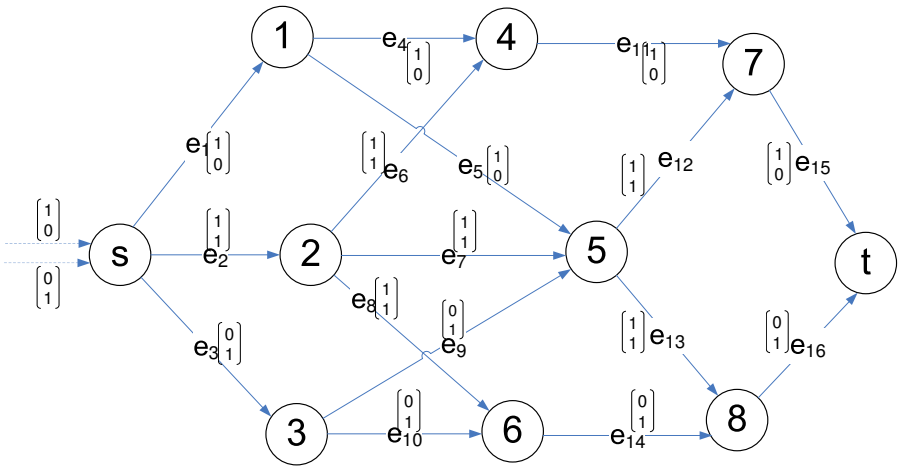

Figure 2.2: Independent set and associated flow

For a linear network code defined on the acyclic network, if the corresponding global encoding kernels of an independent set are linearly independent, then we say that this independent set is *regular*. For any collection of edges $a, b, e_1, e_2, ..., e_i$ where $i \geq 0$, if $Head(a) = Tail(b)$ and $\alpha = \{a, e_1, e_2, ..., e_i\}$ and $\beta = \{b, e_1, e_2, ..., e_i\}$ are independent sets, then independent set α is said to support independent set β and we denote it by $\alpha \rightarrow \beta$. The above concepts are illustrated in the following example.

Example 2 *Figure 2.2 shows a single-source linear network code. We observe that edge e_{13} can be traced back to the imaginary channel via the reversed path $P_1 = e_{13}, e_7, e_2, e_{s2}$ and edge e_{14} can be traced back to the imaginary channel via the reversed path $P_2 = e_{14}, e_{10}, e_3, e_{s1}$. These two paths are edge-disjoint. Thus, $\{e_{13}, e_{14}\}$ forms an independent set and $\{P_1, P_2\}$ is an associated flow for this independent set. Here $f_{e_{14}}$ and $f_{e_{13}}$ are linearly independent, and so $\{e_{13}, e_{14}\}$ is a regular independent set.*

Now let us look at edges e_3 and e_{10}. Edge e_3 is the only

upstream edge of edge e_{10} *and any reverse path from edge* e_{10} *to the imaginary channel must also pass through* e_3*. Thus edge* e_3 *and edge* e_{10} *do not form an independent set. We note that edge* e_{13} *can also be traced back to the imaginary channel by path* $P_3 = e_{13}, e_5, e_1, e_{s2}$*, and* P_3 *and* P_2 *are edge-disjoint. Thus,* $\{P_2, P_3\}$ *forms another associated flow for the independent set* $\{e_{13}, e_{14}\}$*. It is not difficult to verify that* e_8 *and* e_{10} *also form an independent set with a unique associated flow.*

Finally, the global encoding kernels of an independent set are not necessarily linearly independent. For example, $\{e_{12}, e_{13}\}$ *is an independent set, but their global encoding kernels are linearly dependent. We observe that both* $\{e_8, e_{10}\}$ *and* $\{e_8, e_3\}$ *are independent sets and* $Head(e_3) = Tail(e_{10})$*. Thus,* $\{e_8, e_3\}$ *supports* $\{e_8, e_{10}\}$*, i.e.* $\{e_8, e_3\} \rightarrow \{e_8, e_{10}\}$.

□ **End of chapter.**

Chapter 3

A Unified Framework

Summary

The concept of generic network codes is reviewed. A condition regarding the linear independence among global encoding kernels is given. Several equivalent definitions of generic network codes are proposed. A unified framework for linear network codes is introduced. Some existing results are simplified.

3.1 Generic Network Codes Revisited

Generic network codes were first introduced in Li et al. [6] as a way to achieve the multicast capacity in a single-source network. A construction algorithm of generic network code is also proposed in that paper. The original definition of generic network codes is reproduced below for convenience.

Definition 1 *An ω-dimensional linear network code on a single-source acyclic communication network is said to be generic if the*

following condition holds for any collection of edges $e_1, e_2, ..., e_m$ *for* $1 \leq m \leq \omega$*:* $V_{Tail(e_k)} \not\subset \langle f_{e_j} : j \neq k \rangle$ *for* $1 \leq k \leq m$ *if and only if the vectors* $f_{e_1}, f_{e_2}, ..., f_{e_m}$ *are linearly independent.*

This definition has several disadvantages. First, it is conceptually difficult to be understood. It was mentioned in [9] that the motivation for generic network codes is to define a linear network code such that every collection of global encoding kernels that can possibly be linearly independent must be linearly independent. However, it is not clear from [9] what it means by a collection of global encoding kernels being possibly linearly independent. One goal of this paper is to establish the connection between linear independence among global encoding kernels and generic network codes. As we will see later, this connection allows a more concrete interpretation of generic network codes.

Second, the original definition of generic network code does not facilitate the verification of a generic network code. As we will see, the alternative definitions we will present enables such a verification to be done more efficiently and intuitively.

In this paper, we seek simple characterization for a set of global encoding kernels to be possibly linearly independent. The lemma below gives the necessary condition for a set of global encoding kernels to be linearly independent.

Lemma 1 *If the global encoding kernels of a collection of edges* $\{e_1, e_2, ..., e_m\}$*, where* $1 \leq m \leq \omega$*, are linearly independent, then each edge is on some path originating from an imaginary channel and these paths are edge-disjoint, namely these edges form an independent set.*

Proof: Consider a collection of edges $\{e_1, e_2, ..., e_m\}, 1 \leq m \leq \omega$, whose global encoding kernels are linearly independent. We

connect $Tail(e_i)$ to a new node t by a new edge e'_i for $1 \leq i \leq m$, respectively and let $f_{e'_i} = f_{e_i}$ for $1 \leq i \leq m$. Consider any cut U between the source s and node t and let E_U be the set of edges across the cut U. We denote by Mincut(s,t) the min-cut between s and t and by Maxflow(s,t) the max-flow between s and t. Then V_t is a linear transformation of $K(E_U)$, where

$$dim(V_t) \leq dim(K(E_U)) \leq |E_U|.$$

It follows that

$$dim(V_t) \leq min_U |E_U| = \text{Mincut(s,t)}.$$

In particular, for the cut U^* between s and t such that $E_{U^*} = \{e'_i : 1 \leq i \leq m\}$, we have

$$m = dim(V_t) \leq \text{Mincut(s,t)} \leq |E_{U^*}| = m.$$

Thus, Maxflow(s,t) = Mincut(s,t) = m by the Max-flow Min-cut theorem and t can always be traced back to imaginary channels by a set of edge-disjoint paths. Changing the last edges in these edge-disjoint paths from e'_i to e_i for $1 \leq i \leq m$, we obtain the desired set of edge-disjoint paths. We can always do so because $Tail(e'_i) = Tail(e_i)$. □

The above lemma says that a collection of global encoding kernels can possibly be linearly independent *only if* their corresponding edges form an independent set. Thus the best linear network code we can hope for in terms of linear independence is the one in which a collection of global encoding kernels are linearly independent whenever the corresponding edges form an independent set. In designing a linear network code, if the global encoding kernels are required to be independent on only

one independent set, it can be achieved by routing alone. This is illustrated by the example in Figure 3.1. For instance, the global encoding kernels of the incoming edges of node 3 and node 4 can be made linearly independent simply by routing the 2 source symbols to node 3 and node 4, respectively.

If the global encoding kernels are required to be linearly independent on multiple independent sets, since these independent sets may couple with each other through their common edges, routing in general will fail to achieve the desired linear independence. This is illustrated in Figure 3.2. Here, independent set 1 consists of three edges, and independent set 2 consists of two edges. If these two independent sets are regular, then $f_{e_{11}} \neq f_{e_{12}}$, because $f_{e_{12}} \neq f_{e_{16}}$ and $f_{e_{11}} = f_{e_{16}}$. If we do not encode at node R_5, then $f_{e_{12}} = f_{e_8}$ implies that $f_{e_7} = f_{e_{11}}$ which in turn implies that $f_{e_{10}} = f_{e_{15}}$. Thus independent set 1 fails to be regular. Because of the coupling between independent set 1 and independent set 2, routing fails to achieves the desired linear independence.

The situation may change if coding is allowed at the intermediate nodes. An interesting question to ask is whether we can always construct a linear network code in which the global encoding kernels of every independent set are linearly independent. The following lemma provides a positive answer to this question.

Lemma 2 *For any collection of independent sets $\mathcal{I}$, there always exists a linear network code such that any independent set in $\mathcal{I}$ is regular provided $q \geq |\mathcal{I}|$, where q is the size of the base field.*

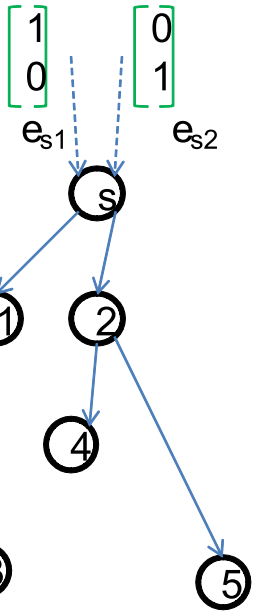

Figure 3.1: For single independent set, linear independence can be achieved by routing alone.

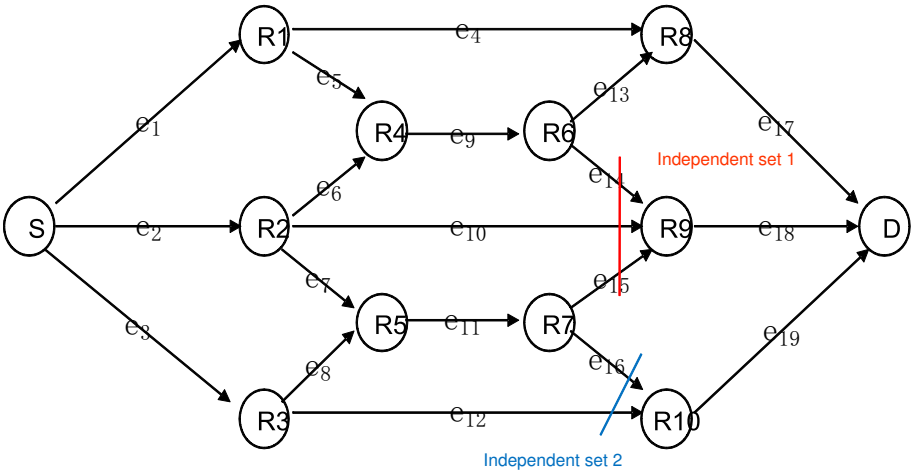

Figure 3.2: Routing in general fails to achieve the desired independence for multiple independent sets.

Proof: We specify the global encoding coding kernels iteratively as in the Jaggi-Sanders algorithm [4]. By definition, each independent set in $\mathcal{I}$ has an associated flow. Initially, only the global encoding kernels of the imaginary channels, namely the standard basis, are specified. In our algorithm, the global encoding kernels are specified in an upstream-to-downstream manner. For each associated flow, the last processed edges on its paths form a *frontier set*. Note that a frontier set is an independent set.

In our construction, we are to maintain each frontier set as a regular independent set. At the beginning, the frontier set of each flow associated with each independent set in $\mathcal{I}$ is a subset of all the imaginary channels. Therefore, each frontier set is a regular independent set to start with. Assume that the regularity of all the frontier sets are maintained at the current step. Let e be the next edge to be processed. Let n be the number of new frontier sets induced by edge e and denote these new frontier sets by $\beta_i, 1 \leq i \leq n$. Suppose $\alpha_i \rightarrow \beta_i$ for $1 \leq i \leq n$, where $\alpha_i, 1 \leq i \leq n$ are the frontier sets in the current step. Denote by $e_i = \alpha_i \backslash \beta_i$ the only edge that belongs to α_i but not β_i and by t the tail of edge e. Since the global encoding kernel of e_i and the global encoding kernels of $\alpha_i \backslash e_i$ are linearly independent for $1 \leq i \leq n$ by the induction assumption and $f_{e_i} \in V_t$ for $1 \leq i \leq n$, $V_t \backslash span(K(\alpha_i \backslash e_i))$ is nonempty. This implies that $dim(V_t \cap span(K(\alpha_i \backslash e_i))) \leq dim(V_t) - 1$ for $1 \leq i \leq n$. If the

base field size $q > |\mathcal{I}| > n$, then we have

$$\begin{aligned}
|V_t \backslash \cup_{1\le i\le n} span(K(\alpha_i \backslash e_i))| &= |V_t| - |V_t \cap [\cup_{1\le i\le n} span(K(\alpha_i \backslash e_i))]| \\
&\ge |V_t| - \sum_{1\le i\le n} |V_t \cap span(K(\alpha_i \backslash e_i))| + 1 \\
&\ge q^{dim(V_t)} - n \times q^{dim(V_t)-1} + 1 \\
&> q^{dim(V_t)} - |\mathcal{I}| \times q^{dim(V_t)-1} + 1 \\
&> 0.
\end{aligned}$$

In the above, the first $\le$ follows from an application of the union bound and the observation that every subspace contains the origin. Thus, by setting the base field size $q \ge |\mathcal{I}|$, we can always choose the global encoding kernel of e to be a vector in $V_t \backslash \cup_{1\le i\le n} span(K(\alpha_i \backslash e_i))$ and the regularity of the new frontier sets can be always maintained. Hence, all the independent sets in $\mathcal{I}$ are regular upon the termination of the algorithm. □

Lemma 1 and Lemma 2 together implies that there exists a linear network code such that the global encoding kernels of a set of edges are linearly independent *if and only if* these edges form an independent set. In other words, the independent set governs the *possibility and impossibility* of linear independence among global encoding kernels. The best linear code in terms of linear independence is the one with every independent set being regular. This coincides with the original motivation of generic network code as explained in [9]. In the following, we prove that a linear network code with every independent set being regular is actually a generic network code. We also prove that a generic network code must have every independent set regular. This gives an equivalent definition of generic network codes.

The original definition of generic network codes has an algebraic interpretation, while the equivalent definition gives a graph-theoretic interpretation which provides more intuition. Another equivalent definition that we will prove in the next theorem gives a simpler way to verify whether a linear network code is generic or not. We only consider the case when $|Out(s)| \geq \omega$, otherwise the problem is degenerate because no node in the network can receive all the information generated at the source node.

Theorem 1 *The following five conditions are equivalent for linear network codes with* $|Out(s)| \geq \omega$.

1. *For any collection of global encoding kernels* $f_{e_1}, f_{e_2} \dots f_{e_m}$, *if* $V_{t_i} \not\subset \langle f_{e_k} : k \neq i \rangle$ *for* $1 \leq i \leq m$ *where* $t_i = Tail(e_i)$ *for* $1 \leq i \leq m$, *then* $f_{e_1}, f_{e_2} \dots f_{e_m}$ *are linearly independent.*

2. *For any collection of global encoding kernels* $f_{e_1}, f_{e_2} \dots f_{e_m}$, *if* $V_{t_m} \not\subset \langle f_{e_1}, f_{e_2} \dots f_{e_{m-1}} \rangle$ *and there exists no directed path from* t_m *to* t_j *for* $1 \leq j \leq m-1$, *where* $t_i = Tail(e_i)$ *for* $1 \leq i \leq m$, *then* $f_{e_m} \notin \langle f_{e_1}, f_{e_2} \dots f_{e_{m-1}} \rangle$.

3. *For any collection of global encoding kernels* $f_{e_1}, f_{e_2} \dots f_{e_m}$, *if* $f_{e_1}, f_{e_2} \dots f_{e_{m-1}}$ *are linearly independent,* $V_{t_m} \not\subset \langle f_{e_1}, f_{e_2} \dots f_{e_{m-1}} \rangle$, *and there exists no directed path from* t_m *to* t_j *for* $1 \leq j \leq m-1$, *where* $t_i = Tail(e_i)$ *for* $1 \leq i \leq m$, *then* $f_{e_m} \notin \langle f_{e_1}, f_{e_2} \dots f_{e_{m-1}} \rangle$.

4. *For any independent set* β, *the global encoding kernels* $K(\beta)$ *are linearly independent.*

5. *For any independent set* α *with* ω *edges, the global encoding kernels* $K(\alpha)$ *are linearly independent.*

Remark: Condition 1 is the original definition of generic network codes [9]. Roughly speaking, Condition 2 means that "new" information must be carried by an edge whenever possible. Conditions 4 and 5 give a graph-theoretical interpretation of a generic network code. They say that if a set of edges can be traced back to the imaginary channels via a set of edge-disjoint paths, then their corresponding global encoding kernels must be linearly independent. Though these five conditions are equivalent, one condition may be more convenient to use than others in different contexts. For example, Condition 4 provides better intuition. Condition 2 is more useful in constructing such a linear network code. Compared with Condition 4, Condition 5 gives a simpler way for us to verify whether a linear network code is generic or not, for we only need to consider independent sets of size ω.

Proof: We will prove that 5)$\Rightarrow$ 4) $\Rightarrow$ 3) $\Rightarrow$ 2) $\Rightarrow$ 1) $\Rightarrow$ 5).

5) $\Rightarrow$ 4): For any independent set β, we can always enlarge it to an independent set α with ω edges by including some edges originating from the source node because $|Out(s)| \geq \omega$. If 5) holds, then the global encoding kernels $K(\alpha)$ are linearly independent. It follows that the global encoding kernels $K(\beta)$ are also linearly independent because β is a subset of α. Thus 5) implies 4).

4) $\Rightarrow$ 3): Let $e_1, e_2, ..., e_m$ be a set of edges such that $f_{e_1}, f_{e_2}, ...,$ $f_{e_{m-1}}$ are linearly independent, $V_{t_m} \not\subset \langle f_{e_j} : j \neq m\rangle$, and there is no directed path from t_m to t_i for $1 \leq i \leq m-1$, where $t_i = Tail(e_i)$ for $1 \leq i \leq m$. We can always find an edge $e'_m \in In(t_m)$ such that $f_{e_1}, f_{e_2}, ..., f_{e_{m-1}}, f_{e'_m}$ are linearly independent, because $V_{t_m} \not\subset \langle f_{e_i} : i \neq m\rangle$. Thus $e_1, e_2, ..., e'_m$ can be

traced back to the imaginary channels via some edge-disjoint paths $P_1, P_2, ..., P'_m$ respectively by Lemma 1. Because there is no directed path from t_m to t_i for $1 \leq i \leq m-1$ and $e_1, e_2, ..., e_m$ are distinct, $P_1, P_2, ..., P_m$, where P_m is the path obtained by appending e_m to P'_m, must also be edge-disjoint paths. Therefore, $e_1, e_2, ..., e_m$ form an independent set. Then $f_{e_1}, f_{e_2}, ..., f_{e_m}$ are linearly independent if 4) holds. Thus 4) $\Rightarrow$ 3).

3) $\Rightarrow$ 2): Suppose a linear network code satisfies 3). Consider any collection of channels $\xi = \{e_1, e_2, ..., e_{m-1}\}$ and any channel $e_m \notin \xi$ such that $V_{t_m} \not\subset \langle f_{e_1}, f_{e_2}, ..., f_{e_{m-1}} \rangle$, where $f_e, e \in \xi$ are not necessarily linearly independent. Then we can always find a subset ξ' of ξ such that $V_\xi = V_{\xi'}$ and $f_e, e \in \xi'$ are linearly independent. Since the linear network code satisfies 3), we have

$$f_{e_m} \notin V_{\xi'} = V_\xi,$$

so this linear network code also satisfies 2).

2) $\Rightarrow$ 1): We prove this by induction on m, the number of edges.

a) Let us consider the case $m = 2$. Assume 2) holds and consider any collection of global encoding kernels $\{f_{e_1}, f_{e_2}\}$ which satisfy 2). Suppose 2) does not imply 1). Then there must exist a directed path from t_1 to t_2. Otherwise, f_{e_1} and f_{e_2} would be linearly independent if 2) holds. Similarly, there must exist a directed path from t_2 to t_1. But this contradicts the fact that the network is acyclic. Thus our assumption is false, and so 2) implies 1) for $m = 2$.

b) Assume 2) $\Rightarrow$ 1) for $m \leq k$ for some $k \geq 2$. We need to show that 2) $\Rightarrow$ 1) for $m = k + 1$. Consider global encoding kernels $f_{e_1}, f_{e_2} \ldots f_{e_{k+1}}$ such that $V_{t_i} \not\subset \langle f_{e_k} : k \neq i \rangle$ for $1 \leq i \leq k+1$. Assume 2) holds. Denote by $\overline{j}$ the set $\{i : 1 \leq i \leq k+1$ and

$i \neq j\}$ for any $1 \leq j \leq k+1$. We observe that $V_{t_l} \not\subset \langle f_i : i \neq l\rangle$ and $\langle f_i : i \in \overline{j}$ and $i \neq l\rangle \subseteq \langle f_i : i \neq l\rangle$ for $1 \leq j \leq k+1$ and $l \in \overline{j}$ implies $V_{t_l} \not\subset \langle f_i : i \in \overline{j}$ and $i \neq l\rangle$ for $1 \leq j \leq k+1$ and $l \in \overline{j}$. By the induction hypothesis that 2) implies 1), global encoding kernels $\{f_{e_i} : i \in \overline{j}\}$ are linearly independent for $1 \leq j \leq k+1$. If 2) does not imply 1) for $m = k+1$, then, for $\forall 1 \leq i \leq k+1$, there must exist a directed path from t_i to some t_j where $1 \leq j \leq k+1$ and $i \neq j$. Otherwise, by 2), $\{f_{e_i} : 1 \leq i \leq k+1\}$ would be linearly independent, a contradiction to that 1) does not hold for $m = k+1$. Since $k+1$ is a finite number, such directed path would produce a cycle which is a contradiction to the assumption that the network is acyclic. Thus, 2) implies 1) for $m = k+1$.

1)$\Rightarrow$ 5): Let $\alpha = \{e_1, e_2, ..., e_\omega\}$ be a size ω independent set. Then there exist ω edge-disjoint paths $P_1, P_2, ..., P_\omega$ from source node s to the channels in α, where the last channel on path P_i is e_i. Denote the length of P_i by l_i and let

$$L = \sum_{i=1}^{\omega} l_i$$

be the total length of all the paths. We will prove the assertion by induction on L. For $L = \omega$, it is easy to check that 1) implies 5), because $Tail(e_i) = s$ for $1 \leq i \leq \omega$ and $dim(V_s) = \omega$. Suppose $K(\alpha)$ is linearly independent for any α with $\omega \leq L \leq k$. We will prove that $K(\alpha)$ is linearly independent for any α with $L = k+1$. Let $A = \{i : l_i > 1\}$ and $\alpha_i = \{e_1, e_2, ..., e_{i-1}, e'_i, e_{i+1}, ..., e_\omega\}$ for $i \in A$, where $e'_i \in P_i$ and $Head(e'_i) = Tail(e_i)$. Then, for α_i where $i \in A$, the global encoding kernels $K(\alpha_i)$ are linearly independent by the induction hypothesis ,which implies $V_{t_i} \not\subset \langle f_{e_k} : k \neq i\rangle$. Also, for any $1 \leq i \leq \omega$ and $i \notin A$, we have $V_{t_i} = V_s \not\subset \langle f_{e_k} : k \neq i\rangle$. It follows

that $V_{t_i} \not\subset \langle f_{e_k} : k \neq i \rangle$ for all $1 \leq i \leq \omega$. If 1) holds, then the global encoding kernels $K(\alpha)$ are linearly independent and we have finished the induction. Thus 1) implies 5).□

We note that 1) $\Rightarrow$ 5) was previously proved in [2]. The condition that there exists no directed path from t_m to t_j for $1 \leq j \leq m-1$ is essential. Otherwise, the equivalence of various conditions may fail to hold. This is illustrated in Figure 3.3. We can verify that this linear network code is a generic network code. It is not difficult to verify that 4) holds. We observe that $V_{Tail(e_1)} \not\subset \langle f_{e_2} \rangle$, but the global encoding kernel $f_{e_1} \in \langle f_{e_2} \rangle$. Thus, 3) does not hold if we do not impose the constraint that there is no direct path from e_1 to e_2.

It is also interesting to note that from 5), we can construct a generic network code by considering only the independent sets with ω edges. In this case, the required field size is $\binom{|\mathcal{E}|}{\omega}$ where $|\mathcal{E}|$ is the number of edges in the network.

3.2 A Unified Framework

Traditionally, a linear dispersion, a linear broadcast, or a linear multicast is characterized by the dimension of incoming global encoding kernel space associated with certain collections of nodes. For example, for a linear multicast, any non-source node t with $maxflow(t) \geq \omega$ has $dim(V_t) = \omega$. For a linear broadcast, any collection of non-source nodes T has $dim(V_T) = min(maxflow(T), \omega)$. However, this approach, referred to as the node-based approach, does not accurately capture the independence structure of linear network codes. For example, Fig-

Figure 3.3: Graph with directed path

ure 3.4(a) is a generic network code and Figure 3.4(b) is a linear dispersion, but the dimensions of V_t and $V_{t'}$ are the same. Therefore, the node-based approach cannot distinguish between a generic network code and a linear dispersion. However, we notice that these two linear network codes have different regular independent sets. The regular independent sets corresponding to the linear network code in Figure 3.4(a) are

$$\{e_1\}, \{e_2\}, \{e_3\}, \{e_1, e_2\}, \{e_1, e_3\}, \{e_2, e_3\}$$

while the regular independent sets corresponding to the linear network code in Figure 3.4(b) are

$$\{e_1\}, \{e_2\}, \{e_3\}, \{e_1, e_2\}, \{e_1, e_3\}.$$

Also, in the node-based representation, different classes of linear network codes cannot be represented in a unified way. As

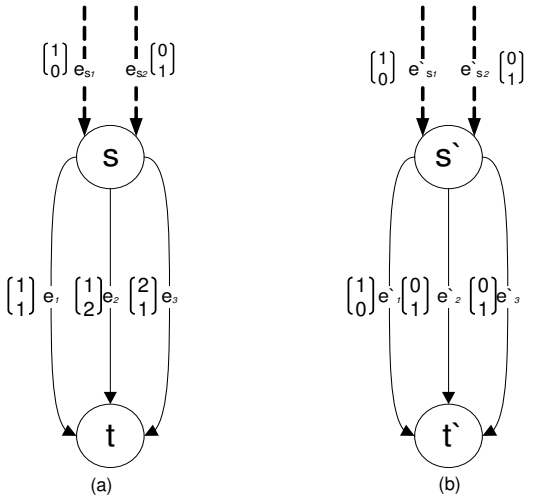

Figure 3.4: Drawbacks of node-based approach

discussed in the last section, in linear network coding, everything boils down to linear independence among global encoding kernels.

We already have obtained necessary and sufficient conditions for a set of global encoding kernels to be linearly independent in Lemma 1 and Theorem 2. Therefore, it is possible that different classes of linear network codes can be represented and constructed in a unified way based on the fundamental concept of linear independence among global encoding kernels.

A unified approach for characterizing different classes of linear network codes based on the concept of linearly independence among global encoding kernels is proposed in this section. All the information regarding linearly independence among global encoding kernels is captured by this framework. Specifically, the

tool of independence set is used to give the "hologram" of linear network codes in terms of linearly independence. We have already seen in Theorem 1 that a generic network code is characterized by regular independent sets. In the rest of this section, we will show that a linear dispersion, a linear broadcast, and a linear multicast can also be characterized by regular independent sets. By using the construction algorithm in Theorem 1, it is not difficult to see that the construction of different classes of linear network codes can also be unified. The original definition of linear dispersion, linear broadcast and linear multicast is reproduced below for convenience.

Definition 2 *[9]A linear network code qualifies as a linear multicast, a linear broadcast, or a linear dispersion respectively, if the following statements hold:*

1. $dim(V_t) = \omega$ *for every non-source node* t *with* $maxflow(t) \geq \omega$.
2. $dim(V_t) = min(\omega, maxflow(t))$ *for every non-source node* t.
3. $dim(V_T) = min(\omega, maxflow(T))$ *for every collection* T *of non-source nodes.*

The lemma below establishes the relationship between linear dispersion and regular independent set and gives an equivalent definition of linear dispersion in terms of regular independent sets.

Lemma 3 (Linear dispersion) *The following two conditions are equivalent for any collection of non-source nodes* T *in a linear network code.*

1. $dim(V_T) = min(maxflow(T), \omega)$.
2. *There exists a size* $min(maxflow(T), \omega)$ *regular independent set* ξ_T *such that* $Head(e) \in T$ *and* $Tail(e) \notin T$ *for any edge* $e \in \xi_T$.

Proof: 1) $\Rightarrow$ 2) : By Lemma 2.27 in [9], we have $dim(\langle f_e : Head(e) \in T, Tail(e) \notin T\rangle) = dim(\langle f_e : Head(e) \in T\rangle) = dim(V_T) = min(maxflow(T), \omega)$. Thus, we can always find a subset ξ_T of $\cup_{t \in T} In(t)$ such that $|\xi_T| = min(maxflow(T), \omega)$ and $\{f_e : e \in \xi_T\}$ are linearly independent.

2) $\Rightarrow$ 1): 2) implies $dim(V_T) \geq min(maxflow(T), \omega)$. Using similar argument as in Lemma 1, we can obtain $dim(V_T) \leq min(maxflow(T), \omega)$. Thus $dim(V_T) = min(maxflow(T), \omega)$, and 2) implies 1). □

In a same manner, we can establish similar results for linear broadcast and linear multicast. The proofs are omitted.

Corollary 1 (Linear broadcast) *The following two conditions are equivalent for any non-source nodes* t *in a linear network code.*

1. $dim(V_t) = min(maxflow(t), \omega)$.
2. *There exists a size* $min(maxflow(t), \omega)$ *regular independent set* I_t *such that* $Head(e) = t$ *for any edge* $e \in I_t$.

Corollary 2 (Linear multicast) *The following two conditions are equivalent for any non-source node* t *in a linear network code.*

1. $dim(V_t) = \omega$ *if* $maxflow(t) \geq \omega$.
2. *There exists a size* ω *regular independent set* I_t *such that* $Head(e) = t$ *for any edge* $e \in I_t$ *if* $maxflow(t) \geq \omega$.

When we specialize $\mathcal{I}$ in Theorem 1 to the corresponding independent sets for linear dispersion, linear broadcast, or linear multicast, we can construct a linear dispersion, a linear broadcast, or a linear multicast, respectively. This gives a unified construction algorithm for linear network codes. From Theorem 1 and Lemma 4, we see that a linear multicast can be constructed provided the field size is larger than $|\mathcal{T}|$ which is the number of receivers. The following example explains the points above.

Example 3 *The linear network code in Figure 3.5 is a linear multicast. We observe that the maxflows of nodes 3, 5 and 6 is at least ω. By Lemma 2, this implies the existence of an associated regular independent set for node 3, 5 and 6 respectively. The associated regular independent set for node 3 is $\{f_{e_3}, f_{e_4}\}$; the associated regular independent set for node 5 is $\{f_{e_6}, f_{e_9}\}$; the associated regular independent set for node 6 is $\{f_{e_5}, f_{e_8}\}$. These three regular independent sets defines a linear multicast.*

3.3 Simplified Proofs

In this section, we will use the insight obtained in last section to provide simplified proofs for some existing results whose original proofs are complicated. It is not difficult to see that a linear dispersion is a linear broadcast and a linear broadcast is a linear multicast. However, it is not obvious that a generic network code is a linear dispersion. The original proof in [9] for this fact is rather complicated. Here we provide a much simpler proof.

Theorem 2 *A generic network code is a linear dispersion.*

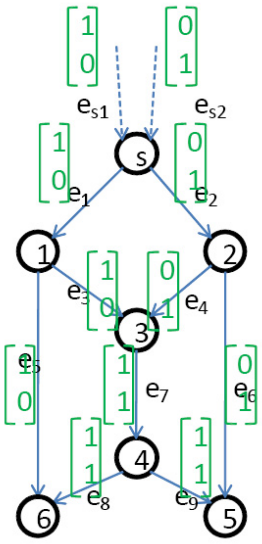

Figure 3.5: Linear multicast and regular independent set

Proof: A generic network code means that all independent sets are regular. In particular, the corresponding independent sets in Lemma 2 are regular. By the definition of linear dispersion, this linear network code is also a linear dispersion. □

For any acyclic graph G, by breaking each edge e_i into two edges e_i^1 and e_i^2 with $Tail(e_i^1) = Tail(e_i)$, $Head(e_i^2) = Head(e_i)$ and $Head(e_i^1) = Tail(e_i^2) = t_i'$ where t_i is a new node inserted in edge e_i, we obtain an extended graph G_E. Figure 3.6 provides one example to illustrate this concept. Now consider any given linear network code defined on the extended graph G_E. Since node t_i' has only one incoming edge, we can assume without loss of generality that $f_{e_i}^1 = f_{e_i}^2$ for all i. Then on the original graph G, by letting $f_{e_i} = f_{e_i}^1 = f_{e_i}^2$ for all i, a linear network code on G is naturally induced by the given linear network code on G_E.

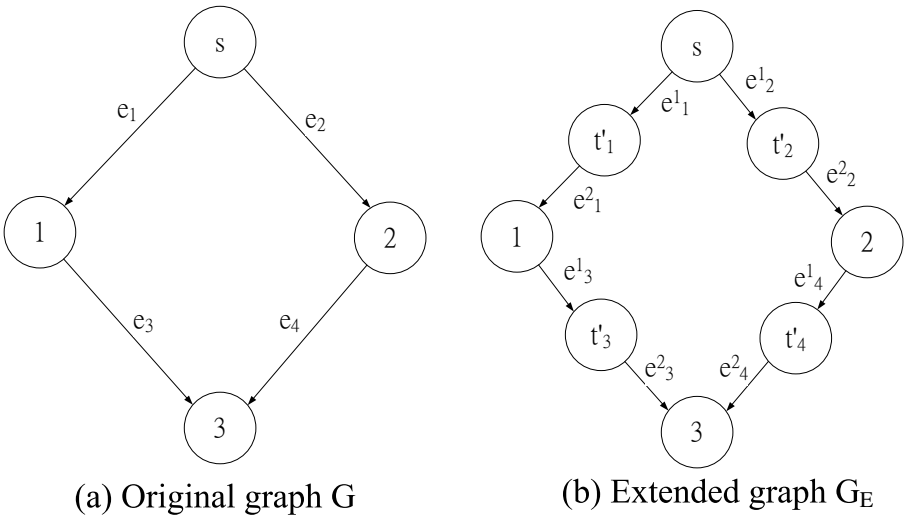

Figure 3.6: Illustration of an extended graph

The following theorem in [5] gives a relationship between generic network codes and linear dispersion defined on the original graph and the extended graph, respectively. Again the proof therein is complicated. A simpler proof based on the unified framework is provided here.

Theorem 3 *Every linear dispersion on the extended graph G_E induces a generic network code on the original graph G.*

Proof: Let G be the original graph, G_E be the extended graph, $\{e_1, e_2, ..., e_m\}$ be any independent set on the original graph, and t'_i be the node inserted in edge e_i for $1 \leq i \leq m$. The incoming and outgoing edges of t'_i are denoted by e_i^1 and e_i^2 respectively. Consider a linear dispersion on the extended graph G_E such that $f_{e_i} = f_{e_i^1} = f_{e_i^2}$ for $1 \leq i \leq m$. This is illustrated in Figure 3.7. The collection of edges $\{e_1, e_2, ..., e_m\}$ is an independent set on G implies that the collection of edges $\{e_1^1, e_2^1, ..., e_m^1\}$

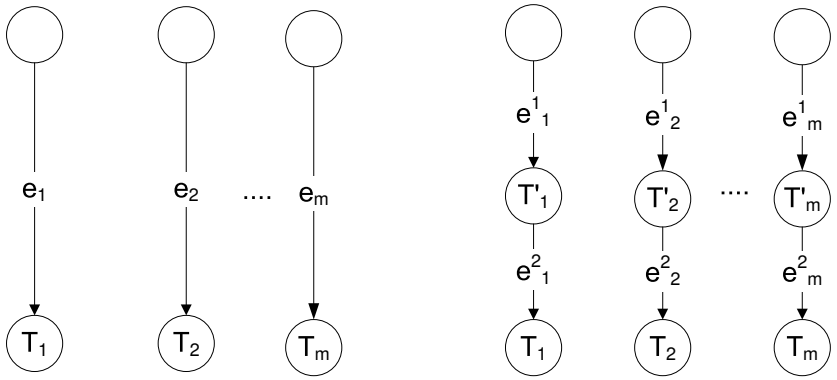

Figure 3.7: A linear dispersion on G_E implies a generic network code on G

is an independent set on G_E. Let $T = \{t'_1, t'_2, \ldots, t'_m\}$. Then $V_T = min(maxflow(T), \omega) = m$ which implies that global encoding kernels $f_{e_1} = f^1_{e_1}, f_{e_2} = f^1_{e_2}, \ldots, f_{e_m} = f^1_{e_m}$ are linearly independent by the definition of linear dispersion. Hence, we conclude that every linear dispersion on the extended graph G_E induces a generic network code on the original graph G. □

□ End of chapter.

Chapter 4

Conclusion

The concept of independence set plays a central role in linear network coding theory. In some sense, it parallels the concept of capacity in classic information theory. In classic information theory, the concept of capacity governs the possibility and impossibility of information transmission, while in the context of linear network coding, the concept of independent set governs the possibility and impossibility of linear independence among global encoding kernels.

In this thesis, the fundamental concept of linear independence among global encoding kernels is studied in depth. Based on this concept, we proved a necessary and sufficient condition for the existence of linear network codes that satisfy certain independence requirement. We proposed and proved the equivalence of several alternative definitions of generic network codes which gives interpretations of generic network codes from different perspectives.

Based on these alternatives definitions of generic network codes, we were able to establish the optimality of generic network codes in terms of linear independence among global encod-

ing kernels. Moreover, we obtained a unified framework for different classes of linear network codes. In particular, this framework suggests a unified construction for such classes of linear network codes.

As applications of our results, we simplified the proofs of some existing results. The results in this thesis can potentially be applied to static network codes and network error-correcting codes.

□ **End of chapter.**

Bibliography

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. 46(4):1204–1216, July 2000.

[2] N. Cai and R. W. Yeung. Secure network coding. In *Proc. IEEE ISIT'02*, June 2002.

[3] T. Ho, B. Leong, M. Medard, R. Koetter, Y. Chang, and M. Effros. The benefits of coding over routing in a randomized setting. In *Proc. IEEE ISIT'03*, June 2003.

[4] S. Jaggi, P. Sandrs, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial time algorithms for multicast network code construction. 51(6):1973 – 1982, June 2005.

[5] P.-W. Kwok and R. W. Yeung. On the relation between linear dispersion and generic network code. 2006.

[6] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. 49(2):371–381, Feb. 2003.

[7] Q.-F. Sun, R. Li, and S.-T. Ho. On network matroids and linear network codes. in preparation.

[8] M. Tan, R. W. Yeung, and S. T. Ho. A unified framework for linear network codes. In *Proc. Netcod'08*, Jan. 2008.

[9] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang. Network coding theory. *Foundation and Trends in Communications and Information Theory*, 2(4 and 5):241–381, 2005.