Nonlinear Network Coding: A Case Study

Lingxiong Li, Kai Fan and Dongyang Long *

Abstract—**In the area of network coding, linear code can achieve the maximum capacity of multicast networks by a large enough alphabet. And it is insufficient for non-multicast networks. Meanwhile nonlinear code is of interest for its possibility to achieve the network coding capacity by small alphabet, or its possibility to deal with other networks. In this paper, a novel polynomial code is proposed. We show that a kind of polynomial code can be induced from linear code, and it has the same coding ability with linear code. We also show the existence of another kind of polynomial code.**

Keywords: network coding, linear code, nonlinear code

1 Introduction

A multicast network is a directed acyclic graph containing a source node, some interior nodes, and a collection of destination nodes. The source node generates messages and transmits them to each destination. A message is a symbol drawn from a fixed alphabet, which is typically a finite field. Each edge of the graph cantransmit one symbol at a time.

In the paradigm of network coding, nodes are allowed to perform both routing and coding operations on the information that they received. A feasible code for a network is the set of operations performed by all nodes that collectively allow destination nodes to receive all messages that their require. Figure. 1 shows such a scheme. The node 3 performs the coding operation and forwards the result along the middle edge. Then, each destination node can compute x and y from symbols they received.

The network coding technique was introduced by Ahlswede et al. [1]. It was proved that a source can multicast k messages to a set of destinations if nodes are allowed to perform coding operations, provided that the min-cut between the source and each destination has capacity k. Li et al. [11] proved that linear network code can achieve the maximum achievable rate for the multicast network, if the alphabet is large enough. Algorithms

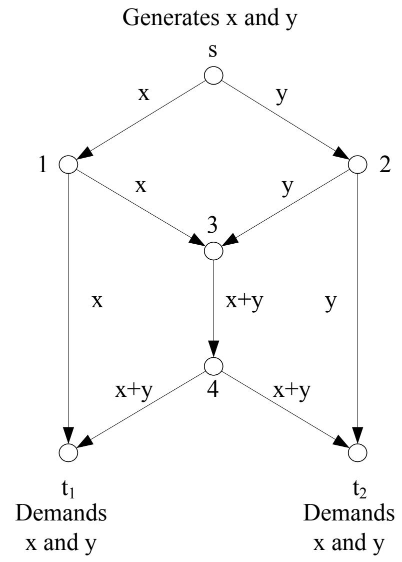

Figure 1: Example of a multicast network with source message x, y, source node s, destination nodes t_1, t_2 and interior nodes 1, 2, 3, 4. x and y are drawn from binary field, + denotes modulo 2 addition .

for constructing linear codes have been studied in [2,7–9]. However, small alphabet was preferred for transmission, since large alphabet cause worse transmission delay and more bandwidth consumption. This motivates the research of decreasing alphabet size. For some networks with special topology, Feder et al. [5] and Rasala Lehman et al. [10] gave the same lower bound of alphabet size, which equal to the square root of the number of destination nodes. It was additionally shown in [10] that the problem of finding the minimum alphabet size for general networks is NP-Hard. Dougherty et al. [3] proved that linear code is no longer sufficient to achieve the maximum capacity of all multicast networks, if the alphabet size is fixed. Meanwhile, nonlinear code may deal with more networks than linear code with the same alphabet resource. For non-multicast network, [4] and [10] showed that linear code is not sufficient, even if the alphabet is allowed to be large enough. Therefore, exploring nonlinear codes was suggested [10] to be a fruitful direction for future works.

To our best knowledge, there are no prior results about constructing a nonlinear code. In this paper, we propose a new schema of nonlinear code, called polynomial code(Section 2). We compare polynomial code with lin-

*Manuscript received August 22, 2007.This work was partially sponsored by the Natural Science Foundation of China (Project No. 60273062, 60573039) and the Guangdong Provincial Natural Science Foundation (Project No. 04205407, 5003350). Lingxiong Li, Kai Fan and Dongyang Long are with the Department of Computer Science, Sun Yat-sen University, Guangzhou 510275, P.R. China. (Tel:86-20-34022899; e-mail:lilx@mail2.sysu.edu.cn) ear code (Section 3.1). The result implies that linear code outperforms polynomial code in a special configuration. While in more general case (Section 3.2), polynomial code is proved to have at least the same coding ability as linear code. We also show how to construct polynomial code from existing linear code. Finally, we show that (Section 3.3) more complex polynomial code exists. Our proof employs an interesting connection between coding function and Latin squares.

2 The Model

We first introduce the necessary concepts and notations. For additional details and definitions, see the references, in particular [1, 9, 11].

A *multicast* problem is a quintuple (G,C,M,s,D):

- A directed acyclic graph $G := (V, E)$.
- Capacities of edges, $C := \{c(e)|e \in E, c(e) \in Z\}$.
- A message of dimension n, $M = (m_1, ..., m_n)$.
- A single source $s \in V$.
- A set of destinations $D \subseteq V$.

The graph G models the network with routers or computers, represented by nodes in V, and communication channels between them, represented by edges in E. The message M was generated at source s, and must be transmitted to all nodes in D. $m_i \in M$ is drawn from an alphabet Σ of size q. The size q is usually assumed to be a power of prime then the alphabet can be a finite field [3]. Without loss of generality, the capacity of edges are usually assumed to be 1 [1]. Thus, each edge can transmit one symbol of the alphabet at a time.

In model of network coding, each edge is associated with an encoding function. It determines the content to be transmitted on the edge when the input message is M. Formally, for edge e from node v to node v', the encoding function ϕ_e can be specified as:

$$\phi_e = \begin{cases} \Sigma^{|M|} \to \Sigma, \text{if } v = s \\ \Sigma^{|E_I(v)|} \to \Sigma, \text{if } v \neq s \end{cases}, \tag{1}$$

where $|E_I(v)|$ is the set of input edges of node v.

A *network code* is defined as the set of encoding function associated with each edge. It characterizes the transmission in the network. A network code is *linear*, if all the encoding functions are linear functions. Otherwise, it is a *nonlinear*. A code is *feasible* if all destination nodes can recover the message M using the information they received. Or else, it is *infeasible*.

Most previous works are focused on linear encoding function (linear network code), because of its implement simplicity. While in this work, we proposed a more general form of encoding function.

Lemma 2.1. *[6] If Σ is a finite field of size q , then any function $f : \Sigma^k \to \Sigma$ can be uniquely represented by a polynomial with coefficients in the field and with degree in each variable at most $q - 1$.*

For example, all functions of $f : \Sigma^2 \to \Sigma$ (q=2) are $\{0, x, y, x+y, xy, x+xy, y+xy, x+y+xy, 0, x+1, y+1, x+y+1, xy+1, x+xy+1, y+xy+1, x+y+xy+1\}$. Since each nonlinear encoding function can be represented by a polynomial with degree greater than one, We denote nonlinear network code as *polynomial code*.

Two functions $f, f' : \Sigma^2 \to \Sigma$ are *independent*, if and only if there exists a function mapping Σ^2 to Σ^2 such that $g(f(\alpha, \beta), f'(\alpha, \beta)) = (\alpha, \beta)$ for every $(\alpha, \beta) \in \Sigma^2$. Or equivalently, if and only if for distinct points $(\alpha, \beta), (\alpha', \beta') \in \Sigma^2$, $(f(\alpha, \beta), f'(\alpha, \beta)) \neq (f(\alpha', \beta'), f'(\alpha', \beta'))$. In short, the input of two independent functions can be recovered from their outputs.

A set of functions $f_1, ..., f_n$ of the form $f_i : \Sigma^2 \to \Sigma$ is called an *independent set*, if they are pairwise independent. A set is called a *maximal independent set*(MIS), if n is maximal. For a MIS, we have $n = q+1$(by combinatorics). If without further specify, the functions appear in what follows are in the form of $f : \Sigma^2 \to \Sigma$.

3 Main Results

In this section, we focus on transmitting messages of dimension two. Firstly, we show that polynomial code is not as powerful as linear code in binary alphabet. Then, it is proved that polynomial code with the same coding ability can be induced from linear code in large alphabet. More complex polynomial code, where interior nodes perform nonlinear operations, is discussed lastly.

3.1 Binary alphabet ($q = 2$)

In this part, we give the comparison of polynomial code and linear code in binary alphabet. Two propositions are given as follows.

Proposition 3.1. *If functions $f, f' : \Sigma^2 \to \Sigma$ are independent, then they are $q-$to-1 mappings.*

Proposition 3.2. *Every linear function $f = ax + by + c$ is a $q-$to-1 mapping, where $(a, b, c) \in \Sigma^3$,$(a, b) \neq (0, 0)$, and 0 denotes the* zero *of the field.*

Next,by Proposition 3.1 and Proposition 3.2, we have the following lemma:

Lemma 3.1. *When $q = 2$, if two functions f, f' are independent, then they are both linear functions.*

The proof is trivial and omitted.

Theorem 3.1. *There exists multicast network that has feasible linear code, but does not have feasible polynomial code.*

Proof. Denote the multicast network in Fig.1 by G. Obviously, G has a feasible linear code. We will show that there is no feasible polynomial code for G. It's suffice to prove that every feasible code is linear. A code is feasible implies that the source message must be recovered from the symbols carried on edges of every cut. Moreover, there are exactly two edges in each cut of G. Thus, if a code of G is feasible, then the functions associated with every cut must be independent. By Lemma 3.1, we conclude that all functions must be linear. Therefore, every feasible code of G is linear. $\square$

3.2 Large alphabet ($q \geq 3$)

The previous part focuses on the binary alphabet. While in large alphabet, codes have a good structure (Lemma 3.2). Although nonlinear code is often considered to be difficult to buildTheorem 3.2 shows that a kind of polynomial code can be easily constructed from linear code.

We introduce the following two propositions:

Proposition 3.3. *Two functions f, f' are independent, if and only if $af + b, f'$ are independent, where $a, b \in \Sigma$, and $a \neq 0$.*

We define the *equivalence set* $[f]$ as $[f] = \{af + b, a, b \in \Sigma, a \neq 0\}$.

Proposition 3.4. *If functions f, f' are independent, then f, g are independent, where $g = af + bf' + c, a \neq 0, b \neq 0, a, b, c \in \Sigma$.*

By Proposition 3.3 and Proposition 3.4, we obtain:

Lemma 3.2. *If functions f, f' are independent, then selecting a function from the set $[f], [f'], [f + f'], ..., [f + (q-1)f']$ will form a MIS, where 1 is the* identity *of the field.*

Let $\pi_i(x_1, ..., x_n) = x_i$, $f_x = \pi_1(x, y)$, $f_y = \pi_2(x, y)$. Selecting a function from each of the set $[f_x], [f_y], [f_x + f_y], ..., [f_x + (q-1)f_y]$ can form a MIS. Let $L = [f_x] \cup [f_y] \cup [f_x + f_y], ..., \cup [f_x + (q-1)f_y]$. Note that $|L| = (q+1)q(q-1)$, thus L contains all linear functions. Specially, we have:

Lemma 3.3. *If $f, f' : \Sigma^2 \to \Sigma$ are linear functions, then they are independent or $f' \in [f]$.*

Next, we get the following result:

Theorem 3.2. *When $q \geq 3$, polynomial code can be constructed from linear code*

Proof. Let f be a polynomial function independent of f_x, then replace f_y by f for all functions of L, we obtain a set $L' = [f_x] \cup [f] \cup [f_x + f], ..., \cup [f_x + (q-1)f]$. Similarly, by replacing f_y by f for all functions of a linear code C, we can obtain a polynomial code C'. If two functions in C are independent, then their counterparts in C' must belong to different equivalence sets of L'. As a result their counterparts in L' are also independent. Therefore, if C is feasible, then C' is feasible too.

Moreover, f does exist: there are $(q!)^q$ functions independent of f_x, while $q^2(q-1)$ linear functions independent of f_x. When $q \geq 3$, we have $(q!)^q > q^2(q-1)$ $\square$

For example, Fig. 2 shows the polynomial code corresponding to Fig. 1.

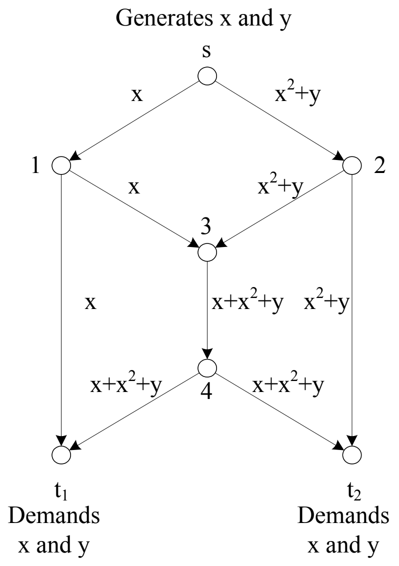

Figure 2: An feasible polynomial code for $q \geq 3$. Source node performs nonlinear operation.

Furthermore, a function f independent of f_x can be easily obtained:

Theorem 3.3. *If function f is in the form of $f = yf_1(x) + f_2(x)$, where $f_1(x), f_2(x)$ are polynomials of x with degree less than q, and for any $x \in \Sigma, f_1(x) \neq 0$, then f is independent of f_x. Especially, the converse holds when $q = 3$.*

Proof. For distinct points $(\alpha, \beta), (\alpha', \beta') \in \Sigma^2$, if $\alpha = \alpha'$, then $f(\alpha, \beta) = \beta f_1(\alpha) + f_2(\alpha)$ and $f(\alpha, \beta') = \beta' f_1(\alpha) + f_2(\alpha)$. since $f_1(\alpha) \neq 0$, we have $f(\alpha, \beta) \neq f(\alpha, \beta')$. Otherwise we have $f_x(\alpha, \beta) \neq f_x(\alpha', \beta')$. Thus, $(f_x(\alpha, \beta), f(\alpha, \beta)) \neq (f_x(\alpha', \beta'), f(\alpha', \beta'))$. Therefore, they are independent.

For the converse: there are $(q-1)^q q^q$ functions in the form of f. While there are exactly $(q!)^q$ functions independent of f_x. Two quantities coincide when $q = 3$, thus the converse holds. □

3.3 More complex polynomial code exists

The previous part shows one kind of polynomial code exist, where only the source node performs nonlinear operations. Does the other kind of polynomial code exist, where interior nodes perform nonlinear operations? An positive example is given in this part.

We introduce some definitions first. A *latin square* [12] of order q is an $q \times q$ square matrix, each row and column of which is a permutation of q different symbols. A function $f : \Sigma^2 \to \Sigma$ can be reformed into a matrix $\{a_{xy}\}$ with its element a_{xy} takes the value of $f(x, y)$. In this way, f_x (f_y) can be written as a matrix with element $a_{xy} = x$ $(a_{xy} = y)$. Therefore, a latin square can uniquely represent a function f that independent of both f_x and f_y.

We denote the total number of different latin squares of order q by $LS(q)$. The value of $LS(q)$ satisfies [12]:

$$LS(q) \geq \frac{(q!)^{2q}}{q^{q^2}} \ . \tag{2}$$

There are $q(q-1)^2$ linear functions independent of both f_x and f_y. We can prove the following.

$$\frac{(q!)^{2q}}{q^{q^2}} > q(q-1)^2, when\ q \geq 5 \ . \tag{3}$$

The detail of proof is omitted to keep concise. Thus, we conclude that there exist polynomial functions that independent of both f_x and f_y. For example, when $q = 5$, $x^3 + y^3$ is such a function. We can construct a polynomial code as shown in Fig. 3.

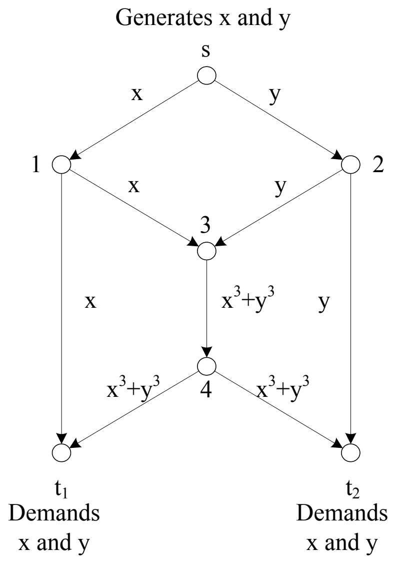

Figure 3: Another feasible polynomial code for $q = 5$. Interior node performs nonlinear operation.

4 Conclusion

In this work, a novel schema of nonlinear code (called polynomial code)is proposed. One kind of polynomial code has the same structure with linear code and can be constructed from them. The other kind of polynomial code is more difficult to construct. The result would be generalized to the configuration of more messages. Our work reveals that the possibility for nonlinear code to outperform linear code lies in the second kind of polynomial code.

References

[1] R. Ahlswede, N. Cai, S. Li, and R. Yeung., "Network information flow," *IEEE Transactions on Information Theory*, V46, N4, pp. 1204–1216, 7/00.

[2] P. Chou, Y. Wu, and K. Jain., "Practical network coding," *Allerton Conference on Communication, Control, and Computing*, 2003.

[3] R. Dougherty, C. Freiling, and K. Zeger., "Linearity and solvability in multicast networks," *IEEE Transactions on Information Theory*, V50, N10, pp. 2243–2256, 10/04.

[4] R. Dougherty, C. Freiling, and K. Zeger., "Insufficiency of linear coding in network information flow," *IEEE Transactions on Information Theory*, V51, N8, pp. 2745–2759, 8/05.

[5] M. Feder, D. Ron, and A. Tavory., "Bounds on linear codes for network multicast," *Electronic Colloquium on Computational Complexity (ECCC)*, V10, N033,2003.

[6] G. Greuel, G. Greuel, and G. Pfister., *A Singular Introduction to Commutative Algebra: Introduction to Communicative Algebra*, Springer, 2002.

[7] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros., "The benefits of coding over routing in a randomized setting," *IEEE International Symposium on Information Theory (ISIT)*, 2003.

[8] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen., "Polynomial time algorithms for multicast network code construction," *IEEE Transactions on Information Theory*, V51, N6, pp. 1973–1982, 6/05.

[9] R. Koetter and M. Médard., "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking (TON)*, V11, N5, pp. 782–795, 10/03.

[10] A. Lehman and E. Lehman., "Complexity classification of network information flow problems," *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 142–150, 2004.

[11] S. Li and R. Cai., "Linear network coding," *IEEE Transactions on Information Theory*, V49, N2, pp. 371–381, 2/03.

[12] J. van Lint and R. Wilson.,*A Course in Combinatorics*, 2nd Edition, Cambridge University Press, 2001.