Int. J. Electron. Commun. (AEÜ) 67 (2013) 1072–1078

Contents lists available at ScienceDirect

International Journal of Electronics and Communications (AEÜ)

journal homepage: www.elsevier.com/locate/aeue

An efficient basic convolutional network code construction algorithm on cyclic networks

Xubo Zhao [a,b,*]

[a] The State Key Laboratory of Integrated Services Networks (ISN), Xidian University, Xi'an 710071, PR China
[b] The College of Science, China University of Petroleum, Qingdao 266580, PR China

ARTICLE INFO

Article history:
Received 23 November 2012
Accepted 24 June 2013

Keywords:
Cyclic network
Basic convolutional network coding
Partial encoding kernel
Construction algorithm

ABSTRACT

Similar to acyclic networks, over cyclic networks, there also exist four classes of optimal convolutional network codes, which are referred to as basic convolutional network code (BCNC), convolutional dispersion (CD), convolutional broadcast (CB), and convolutional multicast (CM), respectively. And from the perspective of linear independence among the global encoding kernels (GEKs), BCNC is with the best strength. In this paper, we present an efficient construction algorithm for BCNC over cyclic networks. Our algorithm can positively provide the maximal required cardinality of the local encoding kernels (LEKs). Another advantage of this algorithm is that for an existing code, when some non-source nodes and associated edges are added, our algorithm can correspondingly modify the already assigned LEKs in a localized manner. And we can just reset the LEKs along some special flow paths educed by the added nodes and edges, rather than reconstructing the whole code in its expanding network.

1. Introduction

We call a directed network cyclic if it contains at least one directed cycle, otherwise, we call it acyclic. Network coding [1] over a cyclic network is essentially different from it on an acyclic one, since there does not exist a partial order on nodes over a cyclic network. Due to the bidirectional communications, in practical setting, most networks are cyclic. The convolutional network code (CNC) can be employed to deal with the propagation of a message pipeline through a cyclic network [2,3]. Corresponding to the optimal classes of linear network codes on an acyclic network [2–4], over a cyclic network, by employing the linear independence among the global encoding kernels (GEKs), Li et al. proposed four types of optimal CNCs at hierarchical levels of strength [5], that is basic convolutional network code (BCNC), convolutional dispersion (CD), convolutional broadcast (CB), and convolutional multicast (CM), respectively. And the existence of BCNC implies that the others also exist, but not vice versa.

It is a significative work to investigate the constructions of optimal CNCs. So far, there have lots of attentions on constructing CM. For example, based on [6], Barbero and Ytrehus [7,8] proposed a polynomial-time centralized CM algorithm. In [9–11], Erez and Feder provided an explicit CM construction algorithm with the binary polynomials as encoding coefficients. On the aspect of BCNC, Li et al. proved that there exists a BCNC with encoding coefficients chosen from any large subset of a discrete valuation ring $\mathcal{R}$ [5]. And by choosing appropriate values for indeterminates in $\mathcal{R}$, they can successfully construct a BCNC. Later, in terms of cycles classification defined by Barbero et al., Huang et al. in [12], gave a BCNC construction algorithm. And then in [13], they extended the algorithm for the four optimal levels CNCs. However, the cardinality of the encoding coefficients is not provided in the above BCNC construction algorithms, since their constructions involve an exhaustive search procedure.

Inspired by the CM construction method proposed by Erez and Feder [9–11], we present an efficient construction algorithm for BCNC over cyclic networks. Our construction can give the maximal required cardinality of the encoding coefficients, which provides a definite domain for picking LEKs. And if we add some non-source nodes and associated edges in an existing code, our algorithm can modify the already assigned LEKs in an efficient localized manner in its expanding network.

This paper is organized as follows. In Section 2, the notation and basic concepts about the CNCs are reviewed, moreover, some new rules are introduced for the convenience of our discussions. In Section 3, we present our BCNC construction algorithm over cyclic networks. And we provide the theoretical guarantee for code construction implementation. Meanwhile, an example is given to illustrate our algorithm construction. The conclusion of this paper is included in Section 4. Appendix A provides the proof of our main theorem.

* Correspondence address: The State Key Laboratory of Integrated Services Networks (ISN), Xidian University, Xi'an 710071, PR China. Tel.: +86 18049634693; fax: +86 02988204275.
E-mail addresses: zhaoxubo_2003@163.com, xbzhao@stu.xidian.edu.cn

1434-8411/$ – see front matter © 2013 Elsevier GmbH. All rights reserved.
http://dx.doi.org/10.1016/j.aeue.2013.06.010

2. Notation and related definitions

An error-free and unit capacity communication network can be modeled as a finite directed multigraph $N = (V, E)$, or simply N, where V is the set of nodes and E is the set of edges. For the directed channel $e = (u, v)$ from node u to node v, denote $u = tail(e)$ and $v = head(e)$. Let $In(v)$ ($Out(v)$) be the set of incoming (outgoing) channels to (from) a vertex $v \in V$. We call an ordered pair (e', e) of channels an *adjacent pair* when there exists $v \in V$ with $e' \in In(v)$ and $e \in Out(v)$. Assume a single source node, denoted by s, let ω be the minimal individual min-cut between s and any of sink nodes. Then $In(s)$ can be regarded as a set consisting of ω imaginary channels.

We define the *directed line graph* of $N = (V, E)$ as $\mathcal{N}(V, E)$, or simply $\mathcal{N}$, with vertex set $\mathcal{V} = E$ and edge set $\mathcal{E} = \{(e', e) \in E^2 : head(e') = tail(e)\}$. That is, each vertex of $\mathcal{N}$ represents an edge of N, and two vertices of $\mathcal{N}$ are connected by an edge if and only if their corresponding edges are an adjacent pair in N.

In a CNC over a cyclic network, we will assign an element called the encoding coefficient or local encoding kernel (LEK) to each adjacent pair (e', e). In order to make the code implementable, as shown in [2,3], all the LEKs are in the form of $p(z)/[1 + zq(z)]$, where $p(z)$, $q(z) \in F[z]$ and $F[z]$ denotes the set of polynomials over the finite field F. In [2,3], functions of this form are called *rational power series*, which will be denoted by $F\langle z\rangle$.

Definition 1. [14] An ω-dimensional F-valued CNC on a communication network N with the delay function $\tau(e', e)$ consists of a LEK $l_{e',e}(z) \in z^{\tau(e',e)}F\langle z\rangle$ for every adjacent pair (e', e), where variable z represents a unit time delay and the function $\tau(e', e)$ maps every adjacent pair (e', e) to a number in $\{0, 1, 2, \cdots\}$. The corresponding GEK for every channel e is an ω-dimensional column vector $g_e(z)$ over $F\langle z\rangle$ such that:

(i) $g_e(z) = \sum_{e' \in In(v)} l_{e',e}(z) g_{e'}(z)$, where $e \in Out(v)$,
(ii) The vectors $g_e(z)$ for the imaginary channels $e \in In(s)$ consist of scalar components that form the standard basis of the vector space F^ω.

The graphical representation of LEKs and GEKs in line graph $\mathcal{N}$ can be found in [15].

We denote $V(\eta) = \{g_e : e \in \eta\}$, where η is a set of channels (nodes) in the original network N (the associated line graph $\mathcal{N}$). A set ξ of exactly ω channels (nodes), including possibly imaginary channels (nodes), is called a *basis* of the original network N (the associated line graph $\mathcal{N}$), if there are ω channel (nodes) disjoint paths starting from imaginary channels (nodes) and ending at channels (nodes) in ξ. And we call this set of ω channel (nodes) disjoint paths an *associated flow* for this basis. Note that a basis may have more than one associated flow. We say a basis ξ is *regular* if rank(span$\{V(\xi)\}$) (the rank of the linear span by all GEKs in set $V(\xi)$) is equal to ω.

Definition 2. [5] An ω-dimensional F-valued CNC on network N is said to qualify as a basic convolutional network code (BCNC) if every basis of the network N is regular.

In our construction, we will employ partial encoding kernels (PEKs) to maintain the regularity of every basis. Before giving the definition of PEK, we need the following notation.

Denote the ith basis in N by ξ_i, and its determined associated flow by $\mathcal{P}_i$. And correspondingly, we label the ξ_i and $\mathcal{P}_i$ in the line graph $\mathcal{N}$. As aforementioned, each associated flow $\mathcal{P}_i$ is consisted of ω node-disjoint paths in $\mathcal{N}$, we denote it by $\mathcal{P}_i = \{P_1^i, \cdots, P_k^i, \cdots, P_\omega^i\}$, where P_k^i is the k-th path of $\mathcal{P}_i$, $k \in \{1, 2, \cdots, \omega\}$. And each path is originated from an imaginary node. Denote a set of ω-nodes $O_i = \{e_1^i, \cdots, e_k^i, \cdots, e_\omega^i\}$ for associated flow $\mathcal{P}_i$, where each node e_k^i belongs to a different path P_k^i in $\mathcal{P}_i$. Initially, the set O_i is consisted of ω imaginary nodes, and it will be updated following subsequent processes. We denote A_k^i be the subset of the path P_k^i which is consisted of all nodes following the node $e_k^i \in O_i$ (not including e_k^i). Let B_k^i be the set of LEKs of the edges with tail in A_k^i and head in $\mathcal{N} \setminus A_k^i$. For sake of clarity, we illustrate these concepts in Fig. 1, in which the node d denotes the last node in path P_k^i, the grey vertexes are the nodes in A_k^i, while the white ones are the nodes outgoing from A_k^i, the dashed arrows represent the edges with tails in A_k^i and heads in $\mathcal{N} \setminus A_k^i$, and the LEKs in B_k^i are specified and set to zero.

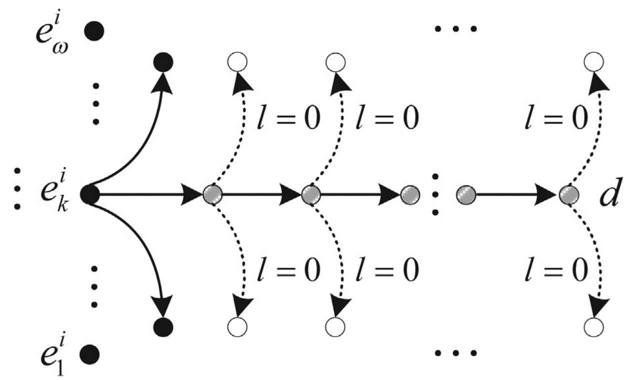

Fig. 1. An illustration of A_k^i and B_k^i in the line graph $\mathcal{N}$.

Definition 3. [11] For any node e in the line graph $\mathcal{N}$, without loss of generality, assume that $e \triangleq e_k^i \in O_i$, then the PEK $u_e(z)$ of the node e satisfies the conditions (i) and (ii) in Definition 1 (with $g_e(z)$, $g_{e'}(z)$ replaced by $u_e(z)$, $u_{e'}(z)$, respectively) when the LEKs in B_k^i are set to zero.

Notice that the difference between GEK $g_e(z)$ and PEK $u_e(z)$ is that for $g_e(z)$, the LEKs in B_k^i may not currently be zero, since they can be determined in previous steps.

3. BCNC construction algorithm over cyclic networks

This section will present our BCNC construction algorithm. Similar to [11] and [16], our algorithm also assumes that the code designer has full knowledge of the network. We know that the propagation delay is an inseparable issue for a CNC over a cyclic network. Thus, before starting our key algorithm, we should introduce suitable delays over the cyclic network in order to make the code causal and implementable. To achieve this goal, we can make use of a precoding algorithm stated in [17], such that we can choose a set of edges denoted by E_N and if we remove them from the network N, there will be no directed cycles. The nodes corresponding to E_N in the associated line graph $\mathcal{N}$ are denoted by $E_\mathcal{N}$. We will assign a delay factor z on every edge (node) in E_N ($E_\mathcal{N}$).

For convenience, in our algorithm, we take the LEKs from a set of binary polynomials. And it is a subset of $\widetilde{F\langle z\rangle}$. Theorem 1 in Section 3.2 can ensure that choosing LEKs in $\widetilde{F[z]}$, the set of binary polynomials with the minimal degree 0 and the maximal degree $\lceil log(C_{n+\omega}^{\omega})\rceil$, is sufficient to maintain the regularities of all bases, where n is the total number of edges (excluding the ω imaginary channels) in N.

3.1. BCNC construction algorithm

Assume that precoding procedure has been done before our construction. The BCNC construction algorithm is shown in the following Algorithm 1 (the subroutine Pick a new LEK is provided in Algorithm 2).