Int. J. Electron. Commun. (AEÜ) 79 (2017) 9–15

Contents lists available at ScienceDirect

International Journal of Electronics and Communications (AEÜ)

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

Regular paper

Improved online fountain codes based on shaping for left degree distribution

Taiqi Huang, Benshun Yi *

Electronic Information School of Wuhan University, Wuhan 430072, China
Shenzhen Institute of Wuhan University, Shenzhen 518057, China

ARTICLE INFO

Article history:
Received 21 December 2016
Accepted 23 May 2017

Keywords:
Rateless codes
Fountain codes
Degree Distribution
Random graph

ABSTRACT

In this paper, an improved encoding scheme for online fountain codes is proposed with the joint optimization of variable node degree and check node degree is proposed. The coding scheme can be divided into the build-up phase and the completion phase. In the build-up phase, left degree distribution is exploited to guarantee optimal performance phase by modifying the traditional coding scheme of choosing input symbols uniformly at random. A memory-based selecting of the source symbols is employed to decrease the number of connected components, which can thus produce the dimension increasement of the linear subspace of a decoding graph constructed in the build-up phase. The upper bound on coding overhead is also derived from the analysis of random graph theory. Compared with conventional online fountain codes, it can be seen from the simulation results that the proposed scheme can provide significant performance improvement with respect to both coding overhead and feedback cost. Moreover, the lower encoding/decoding complexities may make the proposed scheme more practical in energy-constrained applications such as distributed storage.

1. Introduction

Digital fountain codes [1], also called rateless codes, were proposed by Byers et al. in 1998 [2] for reliable communications over lossy networks. Different from conventional fixed-rate codes [3–4], digital fountain codes can adapt the coding rate according to the channel state information. Once a sufficient number of encoded packets which in aggregate is only slightly longer in length than the given message block have been received at the decoder, the source data can be recovered. Luby Transform (LT) codes [5] and Raptor codes [6] are two efficient and practical realizations of rateless codes. As a new class of forward error correction (FEC) codes, rateless codes which can ensure reliable delivery of files with almost no feedback, have been studied in practical systems such as wireless sensor network [7–8], three dimensional video applications [9] and image transmission applications [10].

Conventional fountain codes that are described above suffer from high decoding latency especially when the code blocks are long [11]. In addition, it has been confirmed that not taking the current state of the decoding into consideration can only achieve suboptimal decoding performance [12–14]. Several researches do consider the adjustment of the fixed rateless coding strategy according to the decoding state. In [15], two different types of feedback were utilized to improve the decoding performance for short data-block length LT codes. Compared with existing works, the method of [15] requires lower coding redundancy but suffers from higher coding complexity. In [16], feedback-based LT codes with nonuniform symbol selection distributions were designed to achieve a high intermediate symbols recovery rate.

In [17], Cassuto et al. proposed a new class of rateless codes named online fountain codes. The encoding process can be divided into the build-up phase and the completion phase. In the build-up phase, a uni-partite graph is constructed by coded symbols with degrees not bigger than 2. In the completion phase, the decoding current state represented by the uni-partite graph is fed back to the transmitter. Then the optimal coding strategy is set to increase the linear subspace dimension of the uni-partite graph based upon this decoding state information. The authors proposed a simplified online fountain coding scheme which was simpler for both implementation and analysis. Compared with the previously known codes [18,19], their work can produce a dramatic improvement on the overhead performance. However, the optimal coding strategy is only set in the completion phase of the simplified online fountain codes. When coded symbols with degree 2 are generated in the build-up phase, the authors did not consider the optimal

* Corresponding author.
E-mail addresses: htq@whu.edu.cn (T. Huang), yibs@whu.edu.cn (B. Yi).

http://dx.doi.org/10.1016/j.aeue.2017.05.035
1434-8411/

$$c_1 = s_1 + s_2, \; c_2 = s_2 + s_3, \; c_3 = s_1 + s_3, \; c_4 = s_4 + s_5, \; c_5 = s_6, \; c_6 = s_7.$$

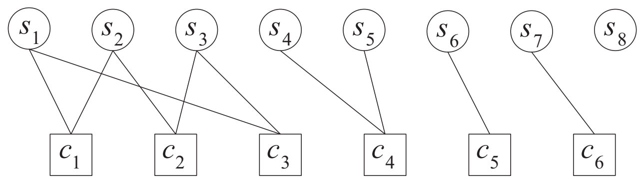

Fig. 1. A bi-partite graph for fountain codes.

way to increase the linear subspace dimension. As a result, the total number of the components is large at the end of the build-up phase, which led to a degradation of performance with respect to coding overhead and feedback cost.

In this work, our focus is on the build-up phase optimization. First, based on the analysis of random graph, we derived the proportionality relation between the number of coded symbols required in the completion phase and the number of connected components constructed in the build-up phase. Thus less coded symbols required to complete the decoding can be achieved if we reduce the connected component number in the build-up phase. Moreover, considering that the size-1 components account for a large portion of the total connected components, we can reduce the coding overhead if the components with size 1 are merged into larger components.

Based on the above analysis, a new online fountain coding scheme is designed. The work in this paper contains contributions in the following respects. First, the proposed optimization method fulfilled with the exploitation of left degree distribution (LDD) enables the increase of the linear subspace dimension. Thus, the optimal performance can also be guaranteed in the build-up phase. On the other hand the proposed generalization of rateless codes gives dramatic improvements in terms of coding overhead and coding/decoding complexities over conventional online fountain codes [15,17]. This might be very important for portable applications such as distributed storage. More XOR operations may exhaust the energy of the energy-constrained storage nodes and reduce the life cycle of the storage system. Finally, performance of the proposed method is compared to other rateless codes with online property through both analysis and simulations.

The remainder of this paper is organized as follows: Section 2 describes the background of online fountain codes. The detail of the proposed scheme and the analysis of upper bound on coding redundancy are described in Section 3. Numerical simulations are performed in Section 4. Finally, we conclude the paper in Section 5.

2. Preliminaries

In [17], the authors address the problem of defining an simplified online fountain code. In this section, we briefly review the simplified online fountain codes introduced by Cassuto.

If the optimal coding strategy can be determined at the transmitter for any given instantaneous decoding state, then this fountain code can be defined as online fountain code. Different from the way of representing the decoding state by the canonical bi-partite graph, the decoding state is denoted by a uni-partite graph. As shown in Fig. 1, the source symbols and the coded symbols are denoted by the circle nodes and the square nodes respectively. The following coded symbols can be depicted as a bi-partite graph detailed in Fig. 1.

The code information can be denoted as a uni-partite graph if the degrees of all the code symbols in the graph are not bigger than 2. The corresponding uni-partite graph which can describe the above coded symbols is depicted in Fig. 2. Connect two source nodes with an edge if the corresponding two source nodes are the "neighbour" nodes of an coded node. A node linked to a coded symbol with degree 1 is coloured black as it can be decoded at the receiver. Otherwise, colour it in white. No edges are connected to the black nodes. For example, s_6 is coloured black as the degree of the code symbol c_5 is 1.

A connected component of a graph is a sub-graph in which any two vertices are connected by edges, and which is not connected to any other vertices in the super-graph. A vertex with no incident edges is itself a connected component. The number of vertices in a component is called the size of the connected component.

Assume that the decoding graph constructed at a given instant contains V nodes which have already been coloured black and V_i size-i connected components. The decoding state then can be denoted by a component enumerator polynomial:

$$V(x) = V + \sum_{i=1}^{k} V_i x^i \tag{1}$$

The number of components is

$$\#component = \sum_{i=1}^{k} V_i = V(1) - V \tag{2}$$

The dimension of linear subspace G established among the coded symbols can be derived from the number of connected components and is given by [17]:

$$\dim(G) = k - \#component(g) \tag{3}$$

Given the decoding state denoted by the component enumerator polynomial, the encoder of the online fountain codes seeks for the maximized elimination probability of a connected component in the graph. Thus by (3) the dimension increases by one. Such increase in dimension can be obtained if a newly degree-d coded symbol can give rise to either of the following two cases:

- Case 1: One single white source symbol and $d-1$ black source symbols are selected to generate a newly received symbol.
- Case 2: Two white source symbols and $d-2$ black source symbols are selected to generate a newly received symbol.

The simplified online fountain code is defined as follows:

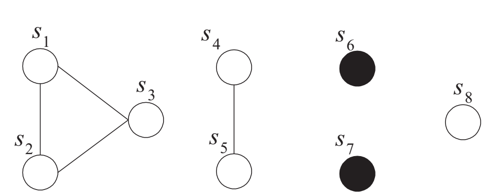

Fig. 2. A uni-partite graph for fountain codes.