Improvement on OFG Decoding Method for LT Codes

J. Guo, S. X. Wu, Z. Q. Wang, X. L. Li
Beijing Information Technology College
Beijing, China

Abstract—**The digital fountain is a network encoding technology of an essentially infinite of encoded symbols, and LT codes is a class of codes that are useful in implementing an ideal digital fountain. Based on analysis of existing LT decoding algorithm, an improved method is proposed. At each encoded symbol arrivals, the new method performs both the triangular and back-substitution steps of Gaussian elimination on the fly, thus distributing the decoding work during all symbols reception and obtaining a short delay.**

Keywords-erasure channel; digital fountain; LT codes; decoding method

I. INTRODUCTION

The erasure channel is a good network-layer model for several packets transmission scenarios. Common techniques to deal with erasures is to use forward error control codes or adopt automatic repeat-request protocols. Recently digital fountain has attracted much attention in the research community [1, 2]. The digital fountain is a network encoding technology of an essentially infinite of encoded symbols, and LT codes is a class of codes that are useful in implementing an ideal digital fountain. In this paper, we investigate decoding schemes for LT codes and show that a proposed decoding scheme improves on the delay to obtain the decoding results.

This paper is organized as follows. Section 2 provides a brief introduction to LT codes. Section 3 then proposes an improved decoding method for LT codes and analyze its performance through simulation. Section 4 gives conclusions of this paper.

II. LT CODES

The basic symbols of LT codes can be packets containing any bits, and the basic symbol operations include symbol copy and bit-wise sum (modulo 2). Assume the source symbols vector is $\mathbf{s}=\left(s_1,s_2,\cdots,s_K\right)$, then the encoded symbols is produced as follows:

Randomly choose the degree d_n of the encoded symbol t_n from a degree distribution $\mu(\cdot)$;

Choose, uniformly at random, d_n distinct source symbols $s_{i_1},s_{i_2},\cdots,s_{i_{d_n}}$, and set $t_n=s_{i_1}\oplus s_{i_2}\oplus\cdots\oplus s_{i_{d_n}}=\mathbf{g}_n\cdot\mathbf{s}$, where $\mathbf{g}_n$ is the coefficient vector of t_n.

This encoding operation defines a graph connecting encoded symbols to source symbols. The decoder need to know the coefficient vector associated with each encoded symbol. When the size of basic symbol is large, this overhead is negligible.

The simple way to decode LT codes is belief propagation (BP). BP decoding is as follows:

Find an encoded symbol t_n that is connected to only one source symbol s_k (if there is no such symbol, this decoding method halts at this point, and fails to recover all the source symbols),

Set $s_k=t_n$,

Add s_k to all encoded symbols $t_{n'}$ that are connected to s_k and remove all the edges connected to the source symbol s_k;

Repeat (1) until all the source symbols are determined.

The probability distribution $\mu(\cdot)$ of the degree is a critical part of LT codes. The objective of designing degree distributions is to meet the following two goals: 1) as few encoded symbols as possible on average are required to ensure complete recovery of the source symbols; 2) the average degree of the encoded symbols is as low as possible. The average degree is the number of symbol operations on average it takes to generate an encoded symbol, and the average degree times the number of encoded symbols is the number of symbol operations on average it takes to recover the entire source symbols. The degree distribution used to form LT codes, the robust soliton distribution, is constructed such that the decoder can recover K source symbols from slightly more than K encoded symbols with high probability. For constants c and δ, the robust soliton distribution $\mu(\cdot)$ is given by

$$\mu(i) = \left[\rho(i) + \tau(i)\right] / \beta$$

where $\beta = \sum_{i=1}^{K}\left[\rho(i) + \tau(i)\right]$ is a normalizing constant, and $\rho(i)$ and $\tau(i)$ are given by

$$\rho(i) = \begin{cases} 1/K, & i = 1 \\ 1/(i+1)i, & 2 \le i \le K \end{cases}$$

$$\tau(i) = \begin{cases} R/iK, & 1 \le i < K/R \\ R\ln(R/\delta)/K, & i = K/R \\ 0, & K/R < i \le K \end{cases}$$

The parameter R represents the average number of degree-one encoded symbols and is defined as

$$R = c\sqrt{K}\ln(K/\delta)$$

Ref. [3, 4] showed that, for a suitably chosen value of c (independent of K and δ), the decoder can recover the source symbols from N encoded symbols with failure probability at most δ, where N is given by $N = K + c \cdot \sqrt{K}\ln^2(K/\delta)$, which goes to K with increasing K.

For erasure channel, Maximum Likelihood (ML) decoding of linear codes is equivalent to solve system of linear equations, where Gaussian elimination (GE) is used for as usual. GE decoding for LT codes requires quadratic-time decoding complexity compared to linear-time decoding complexity of BP decoding. But for small number of source symbols K, GE decoding could be used. There are two efficient versions of GE decoding in practice: Incremental Gaussian Elimination decoding [5] and On the Fly Gaussian Elimination (OFG) decoding [6]. GE decoding consists of two steps: triangular step and back-substitution step. OFG decoding distributes the triangular step during all packets reception, thus obtains a shorter actual decoding time. OFG decoding of LT codes is as follows:

Initialize $K \times K$ coefficient matrix $\mathbf{G}$ and $K \times 1$ encoded symbols vector $\mathbf{t}$;

Upon receiving an encoded symbol t_n, find the position i of the leftmost 1 in its coefficient vector $\mathbf{g}_n$,

If $\mathbf{G}_{ii} = 0$, copy $\mathbf{g}_n$ and t_n into the i th row of $\mathbf{G}$ and $\mathbf{t}$ respectively. Receive t_{n+1}, then repeat step (2);

If the number of 1 in $\mathbf{g}_n$ is less then that in the i th row of $\mathbf{G}$, swap $\mathbf{g}_n$ and t_n with the i th row of $\mathbf{G}$ and $\mathbf{t}$ respectively. Receive t_{n+1}, then repeat step (2);

Otherwise add the i th row of $\mathbf{G}$ and $\mathbf{t}$ to $\mathbf{g}_n$ and t_n respectively. If $\mathbf{g}_n \ne \mathbf{0}$, repeat step (2); else discard $\mathbf{g}_n$ and t_n, then receive t_{n+1} and repeat step (2);

When $\mathbf{G}$ is an upper triangular matrix, solve the source symbols by back-substitution step.

In OFG decoding, the coefficient matrix $\mathbf{G}$ is kept sparse using a swap heuristic in (ii) of step (2), which decrease the computation of the back-substitution step.

III. Improved Decoding Methods for LT Codes

A. *On the Fly BP Decoding Method*

OFG computational effort is distributed on all symbols reception. This feature turns out to be of paramount importance when taking into account the symbols reception delay. In this section, we indicate that BP decoding is also a GE-like method, then we propose on the fly BP decoding and analyze its performance through simulation.

Assume the encoded symbols vector is $\mathbf{t} = (t_1, t_2, \cdots, t_n)^T$, and its coefficient matrix is $\mathbf{G} = (\mathbf{g}_1^T, \mathbf{g}_2^T, \cdots, \mathbf{g}_n^T)^T$. BP decoding of Sec. 2 is described again in matrix form as follows:

Find a row $\mathbf{g}_n$ of coefficient matrix $\mathbf{G}$ that contains only one 1 (if there is no such row, this decoding method halts at this point, and fails to recover all the source symbols),

Set $s_k = t_n$ (suppose the only one 1 is in column k),

All the 1's in column k are canceled and their $t_{n'}$ are added with s_k;

Repeat (1) until all the source symbols are determined.

It can be seen that BP decoding is the back-substitution step of GE decoding, except that $\mathbf{G}$ is not required to be an upper triangular matrix. The robust soliton distribution guarantee that this simple and efficient method can complete decoding.

BP decoding can also work on the fly: upon receiving an encoded symbol, firstly substitute the decoded symbols according to its coefficient vector and modify the coefficient vector; if the coefficient vector contains only one 1, start BP decoding, else receive and process the next encoded symbol.

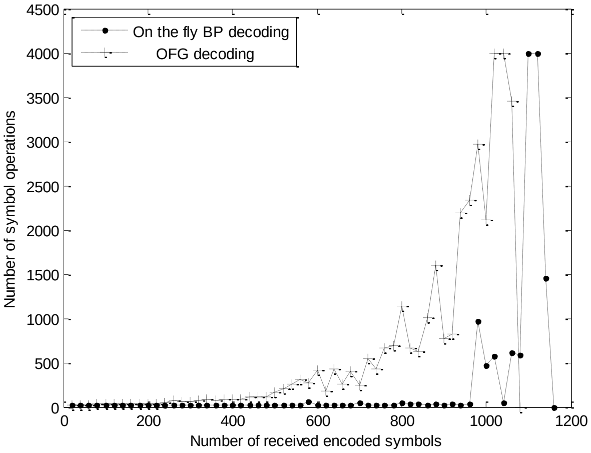

FIGURE I. COMPARISON OF ON THE FLY BP AND OFG DECODING

In Fig. 1, we show the number of symbol operations for on the fly BP decoding and OFG decoding vs the number of received encoded symbols, where the parameters of the LT code are: the number of source symbols $K = 1000$, $c = 0.01$ and $\delta = 0.01$. It can be noted that the computation of OFG decoding goes up with the number of received encoded symbols since the number of empty rows of coefficient matrix $\mathbf{G}$ reduces. When the number of received encoded symbols is larger than 1000, the computation of OFG decoding saturates due to the execution of the back-substitution step (in the simulation, at most 4000 symbol operations can be performed in every symbol interval, which represents the restriction of the decoder's computational ability. The exceeded operations will be performed in the next symbol interval). We also observe that on the fly BP decoding spend almost all the computations after the required number of encoded symbols has been received, as the degree-one encoded symbol of robust soliton distribution is scarce.

B. Improved OFG Decoding Method

OFG decoding performs the triangular step of Gaussian Elimination on the fly, and on the fly BP decoding performs the back-substitution step during all symbols reception. In this section, we propose an improved method by combining both of them, and show simulation results of OFG decoding and the improved one.

The improved OFG decoding works as follows: upon receiving an encoded symbol, firstly substitute the decoded symbols according to its coefficient vector and modify the coefficient vector; then execute OFG decoding; at last, examine the coefficient matrix, if there is an encoded symbol of degree-one, start BP decoding, else receive and process the next encoded symbol.

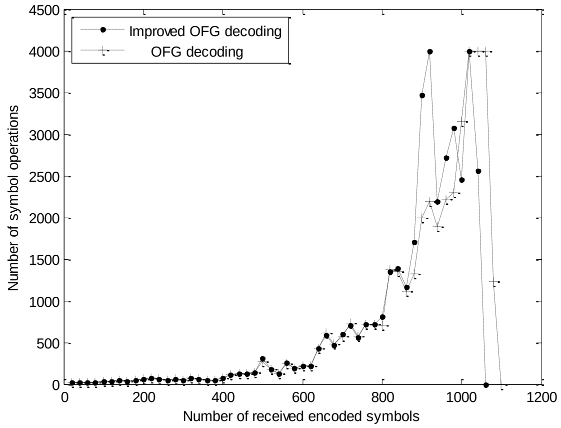

FIGURE II. COMPARISON OF OFG DECODING AND IMPROVED ONE (1)

Fig. 2 shows the number of symbol operations for OFG decoding and the improved one vs the number of received encoded symbols, where the parameters of the LT code are the same as those of Sec. 3.1. It can be noted that improved OFG method distributes the decoding work during all symbols reception. When the number of received encoded symbols is about 900, the computation of the improved method saturates due to the encoded symbol of degree-one appears in the triangular step. Since the improved OFG decoding performs both the triangular and back-substitution steps of Gaussian elimination on the fly, a short delay is obtained.

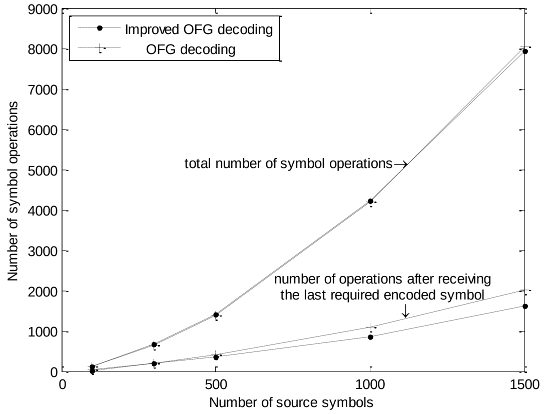

FIGURE III. COMPARISON OF OFG DECODING AND IMPROVED ONE (2)

Fig. 3 shows the total number of symbol operations and the number of operations after receiving the last required encoded symbol in OFG decoding and the improved one for several numbers of source symbols. We present averages of 100 experiments. It can be noted that the total number of symbol operations of both methods are nearly the same, but after receiving the last required encoded symbol, improved OFG decoding reduce the number of operations evidently (about $1/5$ of OFG decoding). Since the new method performs the back-substitution step on the fly, which can reduce the delay to obtain the decoding results.

IV. Conclusions

We have introduced an improved OFG decoding method for LT codes in this paper. This method performs both the triangular and back-substitution steps of Gaussian elimination on the fly, thus distributing the decoding work during all symbols reception and obtaining a short delay.

References

[1] J. W. Byers, L. Michael, M. Michael, R. Ashutosh. "A digital fountain approach to reliable distribution of bulk data", *ACM SIGCOMM Computer Communication Review,* vol. 28, pp. no. 56-67, 1998.

[2] J. W. Byers, L. Michael, M. Michael. "A digital fountain approach to asynchronous reliable multicast", *IEEE Journal on Selected Areas in Communications,* vol. 20, pp. 1528-1540, 2002.

[3] M. Luby. "LT codes", *IEEE Symposium on Foundations of Computer Science,* pp. 271-28, 2002.

[4] D. J. C. MacKay. "Fountain codes", *IEE Proceedings in Communications,* vol. 152, pp. 1062-1068, 2005.

[5] S. Kim, K. Karam, S. Y. Chung. "Incremental Gaussian elimination decoding of raptor codes over BEC", *IEEE communications letters,* vol. 12, pp. 307-309, 2008.

[6] V. Bioglio, G. Marco, G. Rossano, S. Matteo. "On the fly gaussian elimination for LT codes", *IEEE Communications Letters,* vol. 13, pp. 953-955, 2009.