ISSN:0975-9646

Prakhar Srivastava et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (3) , 2012,3970-3974

Digital Fountain in Data Distribution over Non Ad hoc Network

Prakhar Srivastava, Shivam Rao , Vinay Kumar Nirmal
Department of IT, ITM-GIDA
Gorakhpur, India

Abstract— **Digital fountains is the coding technique of data packets before transmitting over network, we can encode packets from the source data and transmit these instead of the source data itself. These packets are then sent across the channel continuously until the receiver has collected enough to decode the file. At receiver end, we can construct all data packets through receiving of only few data packets. The purpose of this paper is to give the good overview about Digital Fountain and how they are used and about their implementation techniques and various algorithms.**

Keywords— **Digital Fountains, LT Codes, Random Linear Codes, Raptor Codes, Encoding, Decoding**

I. INTRODUCTION

Internet connection speeds are now fast enough that people can feasibly download digital audio, stream high definition video, and play real-time online games. Traditionally, large files are transferred across the Internet using protocols such as Hypertext Transfer Protocol (HTTP), and File Transfer Protocol (FTP). These protocols work by splitting a file into equal sized chunks called segments. These segments are then wrapped in packets, which contain the source segments, as well as header information, such as the source and destination ports. Once the segments have been wrapped in packets, the packets are sent across the network. However, on some transmission channels, packets cannot be delivered correctly, or at all. Such channels are known as noisy channels. HTTP and FTP use another protocol called Transmission Control Protocol (TCP), which guarantees delivery of packets. It does this by employing a feedback channel. This works by the receiver sending back acknowledgements of receiving each packet, and unless these are received by the sender, the packets are retransmitted on a noisy channel where the probability of packet loss is high, the number of packets that are sent will also be high, especially when a file is transferred from one server to many clients [3].

This means that the server would have to send roughly twice as many packets as they would if there were no loss, and most clients would have to wait twice as long before they receive the file. There exist, however, many encoding and decoding algorithms which can be used to transmit information correctly across a noisy channel. These work by adding redundant bits to the information which is being transmitted. These redundant bits can be used to detect or even correct transmission errors [3].

II. DIGITAL FOUNTAIN

One of the technique is Digital Fountains, also called as Erasure Codes, are ideal, unlimited data streams injected into the network by one or more source nodes. Suppose that there k numbers of data packets that needs to be transfer, then after encoding of all k data packets, let say n number of encoded symbols are produced and now ready for sending to the receiver, then during transmission if suppose some data packets is lost, then receiver will not get the whole data packets and he will not be able to reconstruct whole file. But using Digital Fountain coding, he can reconstruct the whole file. No matters if some data packets have been lost. This property of Digital Fountain is known as *ideal Digital Fountain code.* This is usually explained with a water fountain analogy, where the encoder is a 'fountain', the receiver is a 'bucket', and the encoded packets are 'droplets' [1] as shown in Fig. 2 and Fig. 3. The way receivers catch data from the unlimited stream resembles the way people quench their thirst at a fountain. No matter which drops of water fall down on earth and which are drunk, the only important thing is that enough drops are caught. We need focus on more generalization of Fountain codes. As it is clear that for any transmission of data packets, first encoding of data packets and then after decoding of data packets is done. In Fig.1 and Fig.4, encoding of k data packet produces n packets. Suppose k' number of packets, that is received for decoding and n-k' packets have been lost. Then there must be a condition which should be followed that is k' must be lies between k and n, otherwise decoding cannot be possible.

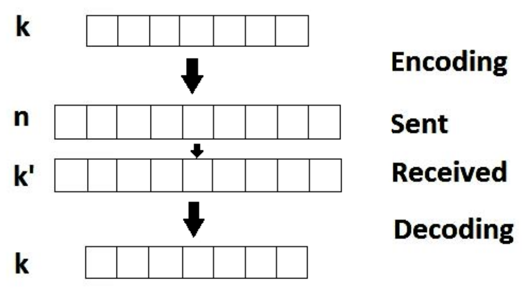

Fig. 1
Digital Fountain (k' must be k'>=k)

Fig. 2 It doesn't matter, what is received or lost

Fig. 3 It only matters that enough is received

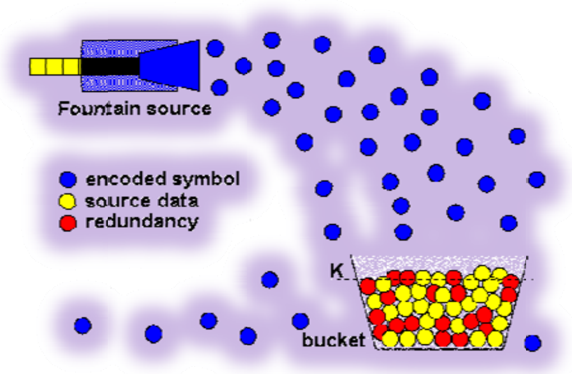

Fig. 4 Generation of Data Packets encode with Fountain Codes

Digital Fountains are rateless because numbers of encoded packets that can be generated from source message are limitless. Fountain codes are universal because they are simultaneously near-optimal for every erasure channel [2]. Regardless of the statistics of the erasure events on the channel, we can send as many encoded packets as are needed in order for the decoder to recover the source data. Encoding and Decoding complexities of Fountain Code are described in [2]. There are several types of Fountain code but in this paper some major types have been focused which are as following:

- Random Linear Fountain
- LT Code
- Raptor Code

III. RANDOM LINEAR FOUNTAIN

In [2], it has been given that if a file size of K packets (composed of whole number of bits) s_1, s_2..........s_k, then at each clock cycle labelled by n, then encoder generates K random bits $\{G_{kn}\}$, and the transmitted packet is equal to the bitwise sum, modulo 2, of the source packets for which G_{nk} is equal to 1. That is:

$$t_n = \sum_{k=1}^{K} s_k G_{kn} \tag{1}$$

Table I
Demonstration of Equation (1)

Let k=3 and n=1

K	S_k	G_{kn}	S_kG_{kn}
1	100111	1	100111
2	001101	0	000000
3	110010	1	110010

Then, $t_n = \sum_{k=1}^{K} s_k G_{kn} = 100111 \oplus 110010 = 010101$

In Table. I, Eq 1 has been demonstrated. In the [1], it has been given that linear code can be represented by matrix vector multiplication of G and x which is y = G. x, where G is encoding matrix, y is the vector of code word corresponding to the vector of word x(Fig. 5).

k columns

$$y = \begin{pmatrix} y_0 \\ y_1 \\ \cdots \\ \cdots \\ \\ \cdots \\ y_{n-1} \end{pmatrix} = G \cdot x = \begin{pmatrix} g_{0,0} & g_{0,1} & \cdots & \cdots & g_{0,k-1} \\ g_{1,0} & g_{1,1} & \cdots & \cdots & g_{1,k-1} \\ \cdots & \cdots & & & \cdots \\ \cdots & \cdots & & & \cdots \\ \\ \cdots & \cdots & & & \cdots \\ g_{n-1,0} & g_{n-1,1} & \cdots & \cdots & g_{n-1,k-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \cdots \\ \cdots \\ x_{k-1} \end{pmatrix},$$

n rows

Fig. 5 Encoding of RL Code, from

Where x= (x_0, x_1, x_2, x_3........x_{k-1}) is vector of k source, and y= (y_1, y_2..........y_{n-1}) is the vector of n code words and G_{nk} is the encoding matrix. Now suppose receiver receives at least k code words out of n code, let y' be the vector of k elements which is received, then:

$$y' = \begin{pmatrix} y_{i,0} \\ y_{j,1} \\ \cdots \\ \cdots \\ y_{l,k-1} \end{pmatrix} = G'x = \begin{pmatrix} g_{i,0} & g_{i,1} & \cdots & \cdots & g_{i,k-1} \\ g_{j,0} & g_{j,1} & \cdots & \cdots & g_{j,k-1} \\ \cdots & \cdots & & & \cdots \\ \cdots & \cdots & & & \cdots \\ g_{l,0} & g_{l,1} & \cdots & \cdots & g_{l,k-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \cdots \\ \cdots \\ x_{k-1} \end{pmatrix}$$

Fig. 6 Decoding of RL Code, from [1]

The encoding matrix G_{kk} can be obtained by extracting those rows from G_{nk} that corresponds to the element of vector y'. For example if y_{j1} is the 2nd element in y', then we have to pick jth row of G_{nk} and inserted as a 2nd row in the matrix G_{kk}. In simple we can say that for decoding we have to calculate linear equation [1]:

Or, $G'. x = y'$

$x = G'^{-1}.y'$

Here receiver must be sure to identify the G_{nk} matrix corresponding to the vector y. Now here a very frequent question arises that what is the probability that G can be reversible or not? The answer and performance of Random Linear code has been shown in [2], by doing experimental results and he calculated that probability comes 0.289 which is not so good. So LT Code should be preferred as described in section IV.

IV. LT CODE

The first idea was invented by Luby in 1998. However in [5], the first practical implementation of this idea was invented by Luby in 2002. The codes he developed were called LT (Luby Transform) Codes, and were practical because the number of packets which we need to receive in order to decode has the upper bound of $K + O(\sqrt{K}(\ln(K/f))^2)$, where K is the number of source segments in the file, and f is the probability of the decoding algorithm failing to complete the decoding. Now suppose we have K=50000 segments and we want to decode with probability 0.9, then upper bound value must be 1700000 packets which good to be practical. . LT codes are rateless, i.e., the number of encoding symbols that can be generated from the data is potentially limitless. Since the decoder can recover the data from nearly the minimal number of encoding symbols possible, this implies that LT codes are near optimal with respect to any erasure channel. Furthermore, the encoding and decoding times are asymptotically very efficient as a function of the data length. Thus, LT codes are universal in the sense that they are simultaneously near optimal for every erasure channel and they are very efficient as the data length grows. For encoding and decoding process of LT Code, first of all there is a need to understand about the bipartite graph and Degree of Distribution. In bipartite graph, degree is the number of edge connected to the node as shown in Fig. 6 and Degree of Distribution has been given in detail in [4] [2] in which *Ideal Soliton Distribution* should be used. Encoding and decoding of LT Code has been given in following section as such given in [2], however Fig. 7, 8 shows the complete demonstration of Encoding and Decoding algorithm as given in [2].

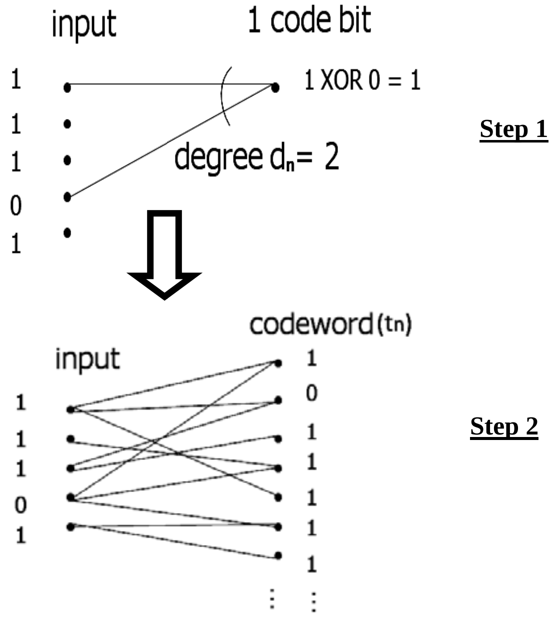

Fig.7 Encoding of LT Code

A. Encoding of LT Code

Each encoded packet t_n is produced from the source file s_1, s_2, s_3...........s_k as follows:

1) Randomly choose the degree d_n of the packet from a degree distribution.
2) Choose, uniformly at random, d_n distinct input packets, and set t_n equal to the bitwise sum modulo 2, of those d_n packets.
3)

B. Decoding of LT Code

1) Find a check node t_n that is connected to only one source packet s_k (Fig. 7). (If there is no such check node, this decoding algorithm halts at this point, and fails to recover all the source packets.)
   - *i.* Set $s_k = t_n$.
   - *ii.* Add s_k to all checks tn, that are connected to s_k
     $t_n := t_{n0} + s_k$ for all n_0 such that $G_{n'k}$ = 1.:
   - *iii.* Remove all the edges connected to the source packet s_k.

2) Repeat (1) until all $\{s_k\}$ are determined.

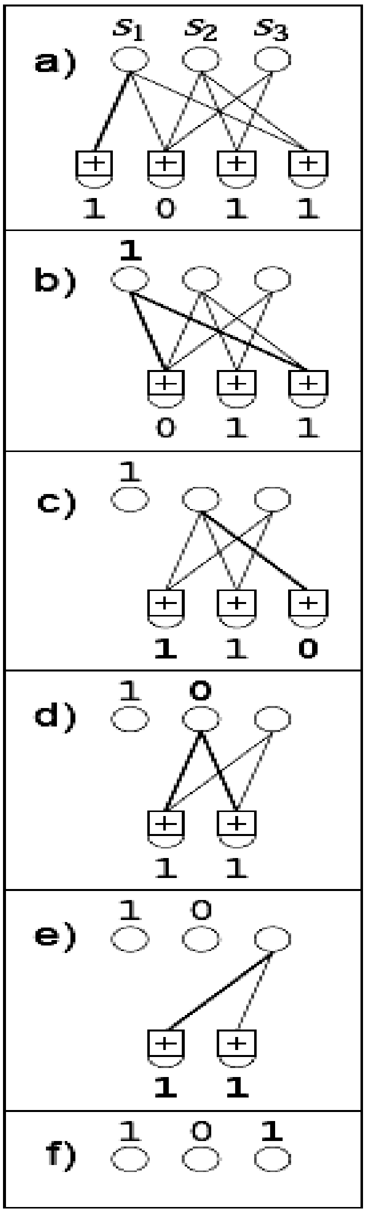

Fig. 7 Decoding of LT Code [3]

V. RAPTOR CODE

Raptor Codes are extension of LT Codes. There are certain limitations of LT Codes [7]. To overcome this problem, Raptor codes had been introduced. The key idea of Raptor Coding is to relax the condition that all input symbols need to be recovered. If an LT code needs to recover only a constant fraction of its input symbols, then its decoding graph need only have O(k) edges, allowing for linear time encoding. We can still recover all input symbols by concatenating a traditional erasure correcting code with an LT code, n intermediate symbols are obtained by encoding k input symbols with an (n, k) erasure correcting block code capable of recovering all input symbols from a fixed fraction of intermediate symbols. The n intermediate symbols are then encoded with an LT code that can recover from its output symbols the required fraction of intermediate A Raptor Code is specified by parameters (k, C, Ω(x)), where C is the (n, k) erasure correcting block code, called the pre-code, and Ω(x)) is the generator polynomial of the degree distribution of the LT code, i.e:

$$\Omega(x) = \sum_{i=1}^{K} \Omega_i x^i \qquad (2)$$

where Ω_i is the probability that the degree of an output node is i. The definition of the encoding cost of a Raptor Code differs slightly - it is the sum of the en- coding cost of the pre-code divided by k, and the encoding cost of the LT code. Raptor Codes also require storage for intermediate symbols, so space consumption is another important performance parameter [7]. In this paper, only introduction part of Raptor code has been discussed. Further summary and probability are given in [1][2][7].

VI. RELATED WORK

In this paper, we are giving in very short about major works that have been done. Luby in 1998 introduced first idea about Fountain code and implemented in 2002. Luby[4] gave the brief introduction and various calculation for LT Code and its Degree of Distribution. However in [2], some modified approach about Degree of distribution has been introduced. Mi-Young Nam [9] gave the generation of encoding matrix of LT Code in contrast. Jonathan Stoten[10] explained various history and explained Shannon theory and hamming codes. Further, now in present scenario, lots of works have been done for optimization of degree of distribution for LT Code and other digital fountains.

VII. CONCLUSION AND FUTURE WORK

In this paper, study of Digital Fountain has been done and overview of Digital Fountain and its major types has been presented. Survey of papers conclude that Luciana Pelusi, Andrea Passarella, and Marco Conti [1] gives the brief description about different types of Fountain Codes, also explained about network coding and introduced about Reed Solomon Codes. Digital Fountain can be used for One-to-Many TCP and point to point Data Transmission. There can be various ways of coding of data packets, but fountain codes are most efficient and it also minimise the network traffic and helps to improve the congestion control by recovering the whole file, through receiving of only few data packets. One of the best scope of Digital Fountain can be that it should be used for Wireless medium communication where no acknowledgement is sent and Satellite Communication where One-to-Many Data transmission is needed with reliability and faster communication.

REFERENCES

[1]. Luciana Pelusi, Andrea Passarella, and Marco Conti, "*Encoding for Efficient Data Distribution in Multi-hop Ad hoc Networks*" available from cnd.iit.cnr.it/andrea/docs/chap_netcod07.pdf

[2]. David J.C. MacKay, Cavendish Laboratory, University of Cambridge, "*Fountain Codes*" available from *www.inference.phy.cam.ac.uk/**mackay/fountain**.pdf*

[3]. David J.C. MacKay **,** Information Theory, Inference, and Learning Algorithms, Cambridge University Press 2003

[4]. M. Luby, "LT codes", *in Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 271-282, 2002.

[5]. Berlekamp, E.R.: "*Algebraic coding theory*" (McGraw-Hill, New York,1968)

[6]. H. Kushwaha and R. Chandramouli, "*Secondary spectrum access with LT codes for delay-constrained applications*" ,[in Proc. IEEE Consum. Commun. Network. Conf., Las Vegas, NV, Jan. 2007]

[7]. Tracey Ho, Summary of Raptor Codes, October 29, 2003

[8]. A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Ubiquitous Access to Distributed Data in Large Scale Sensor Networks through Decentralized Erasure Codes", *in Proceedings of the Fourth International Conference on Information Processing in Sensor Networks (IPSN 2005)*, April 25-27, Sunset Village, UCLA, Los Angeles, CA.

[9]. Mi-Young Nam "*Generation of encoding matrix for LT codes*" YONSEI University June 2009

[10]. Jonathan Stoten "*Digital Fountain Code*"Royal Holloway, University of London, March 2010