Data Distribution in Ad hoc network using Fountain code in OMNeT

Rohit Kumar Gulshan[1]
Computer Science & Engg.
Sharda University, India

Gouri Sankar Mishra[2]
Computer Science & Engg.
Sharda University, India

Abstract - This paper provides an efficient method for delivering data chunks from source to destination in ad hoc network using fountain code. There is a probability of loosing data packets very often. Not only that, in a ubiquitous environment where some or all the nodes are in constant move, we need to make an ad-hoc network which can connect to each other whenever required. Keeping the same in context the concept of rateless erasure codes in ad hoc network has been introduced. There is a challenge of minimizing the communication time and improving efficiency in data transferring between communicating nodes, for that proposed method has been used. Since fountain codes are information additive, i.e. a particular receiver can accumulate encoded packets from multiple sources in order to decode the source information. This is the idea to use fountain codes in ad-hoc networks

I. INTRODUCTION

Fountain codes are relatively new technique, designed for reliable data distribution in lossy and noisy computer channels and networks. Fountain codes are unlike any other conventional error correction codes. To meet the need for fast and reliable information exchange, mobile and communication networks have become an important factor for Ad hoc networks which enhances the communication capabilities by providing connectivity at anytime, anywhere.

In a noisy channel, there is a probability of loosing data packets very often. So to reduce the effect of loss of packets, LT codes [1] are introduced, we propose to exploit LT code, which is one of the fountain codes. Not only that, in a ubiquitous environment where some or all the nodes are in constant move, we need to make an ad-hoc network which can connect to each other whenever required. Suppose there is a string of input symbols $(x_1,\ldots,x_k)$, a LT code produces a limitless stream of output symbols. These output symbols are produced independently by using a technique known as degree distribution D on F_2^K and later on adding the input symbols to it. The symbols generated by a LT code after the degree distribution is called output symbols. Luby invented the class of LT codes with faster encoding and decoding algorithms. LT (Luby transform) codes are special case of digital fountain's codes. LT encoder [1] chooses randomly from a set of *K* packets, and according to a distribution, a subset of *d* packets, and XOR them.

II. FOUNTAIN CODE (LT CODE) WORKING

LT codes are rateless erasure codes. The rateless because we do not have to configure first the probability of channel erasure before transmitting. It is supposed that they have property that infinite numbers of encoded packets "from the fountain source" can be created from a source file or segment.

During the encoding and decoding implementation, it is not considered that packets are lost, the only thing is considered that enough packets has been received by the bucket then original file is constructed from received packets. We called packets as "droplets" after the encoder produced output symbols.

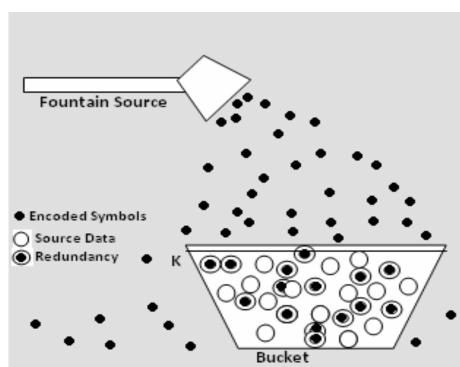

Figure 1. Diagram showing basic functioning of fountain code (LT code)

The transferred data can be then encoded algorithmically by choosing random number of droplets sent over the ad hoc network. The receiver needs to catch some random number of droplets to reconstruct the original file. It doesn't matter which droplets are received or in what order they are received and also which droplet has been lost. If sufficient amount of droplets are received then the original file can be reconstructed. [2][3][4] Algorithm 1 and 2 demonstrates the encoding and decoding procedures respectively.

Algorithm 1: LT Encoding

1. Repeat
2. Choose a degree d from degree distribution *p(d)*
3. Choose uniformly at random d input symbols $n(i_1),..,n(i_d)$.
4. Calculate value $n(i_1)$ XOR $n(i_2)$ XOR ... XOR $n(i_d)$
5. Until stop bit received

Algorithm 2: LT Decoding

1. Repeat
2. If d = 1 packet in buffer
3. n(j) ← recover j
4. For all n(j) in buffer : v includes n(j) do
5. d ← d - 1 (reduce degree)
6. v ← v XOR n(j) (update value)
7. End for
8. Until all input symbols recovered

III. SIMULATION PROCEDURE IN OMNeT

Any simulator can be used for fountain codes. Since the simulation is carried out in OMNeT, the implementation of fountain codes can be done in OMNeT itself. Write the fountain code "code" as a separate "function" and before transmitting the respective data from one node to another, fountain encode it and then send it. Similarly it can have a separate "function" for the decoder. At the receiver end it's needed to call the decoder function and decode the original data.

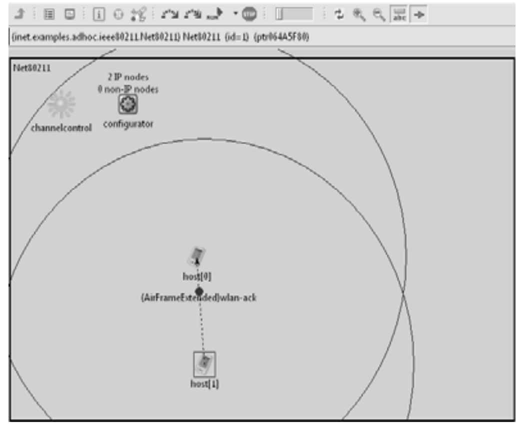

Figure 2. Network for testing the IEEE 802.11b model in ad-hoc mode using mobile nodes in OMNet++

Since there is a proposed ad hoc network, through which data is sent over the network, it can be achieved by supposing, two nodes network which are sending and receiving the data respectively. There is some loss of data during the process of sending and receiving due to lossy nature of network and this is why it can be supposed as an ad hoc network, as the nodes moves within the geographical mean, it is quite natural. The encoding and decoding is done in the physical layer and the sending and receiving of droplets can be realized in transport layer of seven-layer OSI model.

IV. CONCLUSION & FUTURE WORK

It has been found that, after experimenting with the algorithms, the client has to receive about 4 times the amount of data being transferred before being able to reconstruct the required original file (*Figure* 3). The results and implementation is not found to be industry level. But by changing the method of selection of droplets can improve the performance of fountain codes. These changes are strictly based upon the method of catching the droplets and suitable for the real world application. Fountain codes have some downside too when transmitting, in the integrity of the data. If somehow not enough droplets are caught then original file cannot be reconstructed.

Another important factor is the implementing the Robust Soliton distribution for the encoder which is the important part. For the decoder implementation creates a binary generator matrix during the encoding itself and uses that matrix for implementing the decoder.

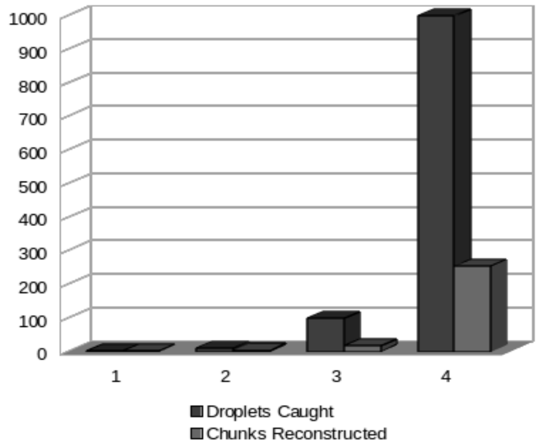

Figure 3. No. of Droplets vs. No. of Chunks.

Client has to receive about 4 times the amount of data being transferred before being able to reconstruct the required original file

Future work will be to optimizing LT code by reducing the number of droplets to decode data at the client side. And to use more improved protocols other than just using Hyper Text Transfer Protocol. It is also to be taken into account that for practical use of fountain codes, it would be better to work with the file at the bit level rather than at the character level.

ACKNOWLEDGEMENT

It is most gratuities to my respected guide Mr. Gouri Sankar Mishra for giving his valuable time and continuous guidance during my research work. I mostly thanks to my friends for their supporting. I also very much thankful to my parents who gave me their parental nourishment and influence me for research work.

REFERENCE

[1] M. Luby, "LT-codes," in *Proceedings of the ACM Symposium on Foundations of Computer Science (FOCS)*, 2002.

[2] D.J.C. MacKay, "Capacity Approaching Codes Design and Implementation", Fountain Codes, IEE Proc-Commun, Vol.152. No. 6, December 2005, doi: 10.1049/ip-com:20050237.

[3] A. Shokrollahi, "Raptor Codes", Foundation and Trends in Communications and Information Theory, IEEE Information Theory Society, doi: 0.1109/TIT.2006.874390.

[4] D.J.C. MacKay, "Information Theory, Inference, and Learning Algorithms", Cambridge University Press,2004.

AUTHORS

Mr. Rohit Kumar Gulshan is a M-tech (CSE) Scholar from Sharda University. He has studied B.Tech in Information Technology from GBTU formally UPTU in 2010. He has interests in Network Security, Ethical Hacking and Information Systems.

Mr. Gouri Sankar Mishra has been joined in Sharda University as an Assistant Professor. He has worked on Network Securities, Network Protocols and Information Systems.