ISSN No:-2456-2165

Implementation of Luby Transform Error Correcting Codes on FPGA

[1]K. L. Sudha, [2]Ganesh Gowda, [3]Gagan K.P, [4]Adarsh Reddy P.
[1]Professor, [2,3,4]UG Students,
Dept.of ECE DSCE, Bengaluru, INDIA

Abstract:- Luby Transform (LT) codes are rate less codes which are a type of fountain codes. They provide good performance over other fountain codes because, more efficient encoding and decoding algorithms can be devised for this code. These codes are rate less codes because they allow for flexible and adaptive rate allocation during the encoding process. With this, any desired code rate can be achieved by controlling the number of parity symbols generated during the encoding process. This paper explains the encoding and decoding aspects of LT codes in detail and generation and decoding is performed using MATLAB. Hardware implementation of LT code is attempted on FPGA by using Verilog code. As it is shown, the codes are very simple to implement and can be used as a more powerful error correcting codes.

Keywords:- *Forward error correction, Rate less codes, Verilog, FPGA.*

I. INTRODUCTION

Forward error correction in wireless communication is one of the important aspects to see that reliable transmission happens over noisy channel. Rate less codes are types of codes which adjust its rate as per the channel conditions and were originally developed to attain efficient transmission in erasure channels. Luby transform (LT) codes are the first practical realization of rate less codes which was suggested by Luby in his paper [1], published in 2002. It is a type of Fountain code. In the fountain codes, the transmitter sends encoded blocks over the channel which alters the number of encoded bits depending on the channel's properties [3]. Raptor codes [2] are the extension of LT CODE.

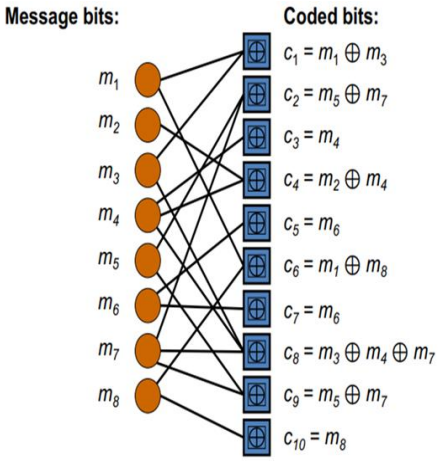
II. LITERATURE REVIEW

M. Luby introduced LT codes in 2002[1] which are rate less codes. He gave encoding and decoding algorithms and showed that they provide a flexible and adaptive mechanism for adjusting the code rate, and can achieve near-optimal error correction performance with a small number of parity symbols. David J.C MacKay[2] proposed an improved LT coded and called it as Raptor code where in an extra layer of channel coding is introduced. Paper [3] studies the performance of two classes of rateless codes-LT and Raptor codes on noisy channels and concludes that Raptor codes outperform LT codes, and have good performance on a wide variety of noisy channels. Authors of paper [4] proposes a modified encoding scheme for LT codes that maximizes the minimum variable-node degree for transmission over binary erasure channels. The proposed scheme leads to an almost-regular variable-node degree distribution. Paper [5] revisits the fountain codes and proposes a research work which uses fountain coding in encoding processes for wireless channel transmissions. An. improved encoding scheme for LT codes is designed in paper [6] by classifying the information nodes according to their degrees to reduce the error floor. Paper by Khaled F [7] proposes memory based LT codes for 5G network and beyond. Fountain codes and their applications in wireless communication system have been dealt in paper [8]. Systematic LT codes have been used as channel coding technique by authors of paper [9] to transfer AES encrypted Image on BPSK over AWGN channel. Paper [10] explains the usage of LT codes for error correction in storage devices and has shown that the performance of Lower Triangular Matrix-LT codes perform better compared to the traditional schemes such as RSD used in Distributed Storage System. Authors of paper[11] have implemented LT codes on Arduino board and NRF transceivers.Maintaining the Integrity of the Specifications

III. LT ENCODING AND DECODING

A. Encoding algorithm:

The LT *code* works by dividing *the original message into a set* of small packets, each of which is independently encoded using a random linear code. This means that each packet is multiplied by a randomly generated matrix of coefficients, resulting in a unique code word for that packet. The coefficients used in this process are chosen from a finite field, such as GF(2). Actually it is a linear block code which is generated with a sparse generator.



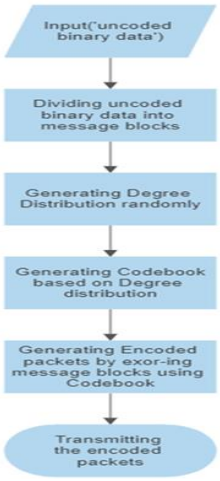


Fig. 1(a): Message partitioning and packet forming Fig. 1(b):flow chart for encoding

The generator matrix for an LT (Luby Transform) code is generated using a random linear code construction. The construction process involves the following steps:

- Message partitioning: The original message to be encoded is divided into a set of packets, each of which is assigned a unique identifier. Red dots in fig(1a) indicates that the sequence of bits are divided in to 8 message blocks.
- Coefficient generation: A set of random coefficients is generated for each packet, using a pseudorandom number generator. The coefficients are typically chosen from a finite field. This is called degree distribution. This is done with codebook creation and generation of random numbers.
- Packet encoding: Each packet is multiplied by its corresponding set of coefficients based on degree distribution, to create a unique code word. The code words for all packets are concatenated to form the generator matrix for the LT code. Code words are shown as blue square boxes in fig(1)

The generator matrix is then used to encode the message into a set of coded packets, which are transmitted to the receiver. At the receiver, the coded packets are decoded using a probabilistic decoding algorithm, which relies on the properties of the generator matrix to recover the original message with high probability.

B. Decoding algorithm for LT code

The decoding algorithm for an LT (Luby Transform) code is a probabilistic algorithm that uses the generator matrix of the code to recover the original message with high probability. The decoding process involves the following steps:

- Packet reception: The receiver collects a set of coded packets transmitted by the sender. Each packet includes an identifier and a set of coefficients.
- Matrix construction: A submatrix of the generator matrix is constructed using the identifiers of the received packets. The submatrix includes only the rows corresponding to the received packets, and its columns correspond to the coefficients used to generate the packets.
- Solving the linear system: The receiver uses the submatrix to solve a linear system of equations, where the unknowns are the message packets. This involves inverting the submatrix (if it is full rank) or solving the system using a linear solver.
- Message reconstruction: Once the linear system is solved, the receiver obtains estimates for the message packets. If the estimates are correct, the original message can be reconstructed by concatenating the packets in their correct order.
- Iterative decoding: If the original message is not correctly reconstructed, the receiver can repeat the decoding process with additional received packets. This involves constructing a new submatrix of the generator matrix that includes the new packets and the rows that were not used in the previous iteration. The linear system is then solved again to obtain improved estimates for the message packets.

The actual decoding process (shown in fig 2a and b) first identifies received packets with degree one, where each of these packets is a symbol. These symbols are placed into a buffer denoted as a ripple. Then, each of these symbols rj is taken out from the ripple to decode all the other packets with a degree more than one. For a packet ci, if it contains rj , an XOR operation is performed between rj and ci which results in reducing the degree of ci by one as the symbol rj is eliminated from packet ci. Moreover, after the XOR operation, if the degree of ci becomes one, then ci is reduced to a symbol. This newly recovered symbol is then moved into the ripple if it is not found in the ripple.

ISSN No:-2456-2165

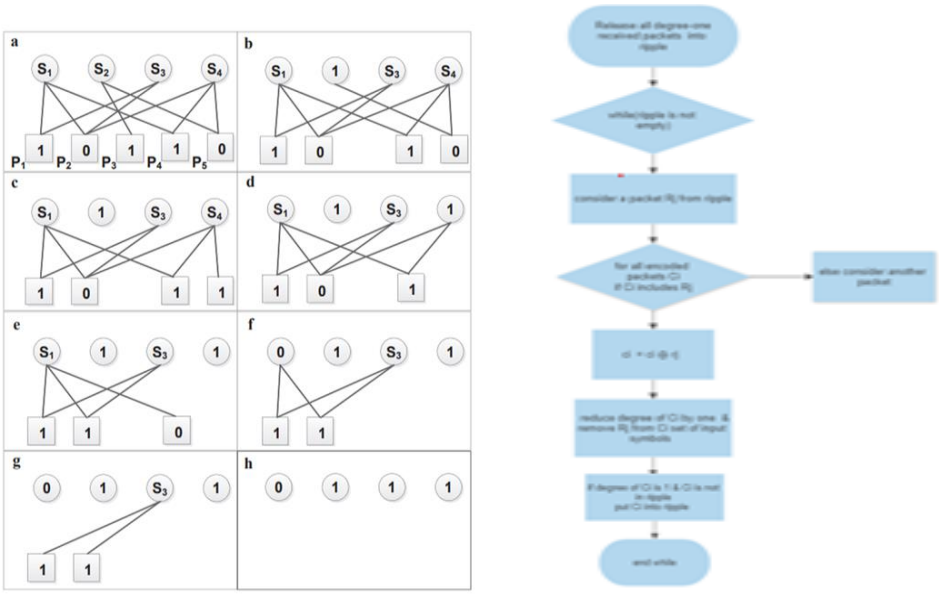
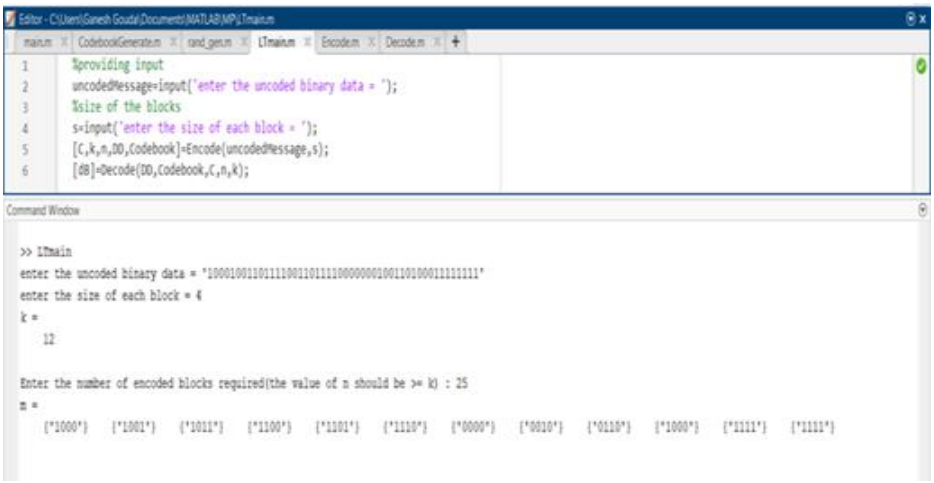


Fig. 2(a): Decoding algorithm with flow chart

The process of taking a symbol out from the ripple for decoding other packets and placing newly recovered symbols into the ripple is repeated until the ripple is exhausted. The ecoding process fails if there is at least one symbol yet to be discovered when the ripple is empty. Otherwise, the process is said to be successful since all symbols are decoded.

The simulation of encoding and decoding of LT code is done in MATLAB simulation tool. Figures 3,4,5 shows the encoding process and decoded messages.

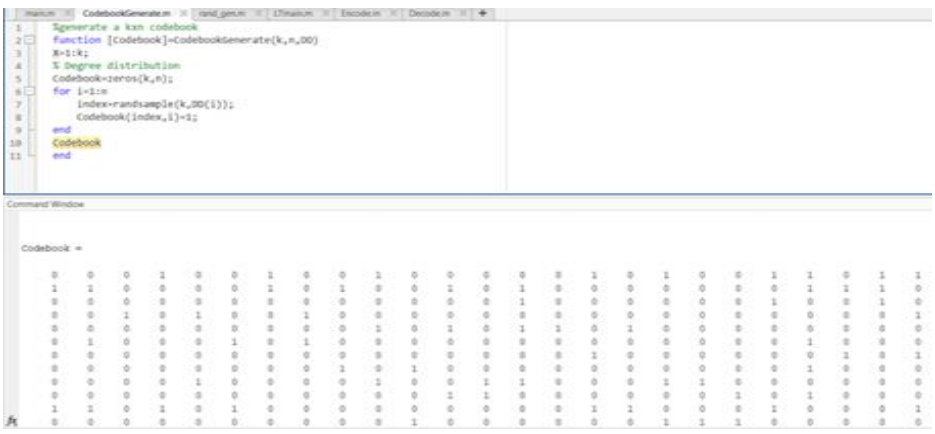


```
%providing input
uncodedMessage=input['enter the uncoded binary data = ');
%size of the blocks
s=input('enter the size of each block = ');
[C,k,n,DD,Codebook]=Encode(uncodedMessage,s);
[dB]=Decode(DD,Codebook,C,n,k);
```

```
>> LTmain
enter the uncoded binary data = '1000100110111100110111100000001001101000111111111'
enter the size of each block = 4
k =
    12

Enter the number of encoded blocks required(the value of n should be >= k) : 25
n =
    ('1000')    ('1001')    ('1011')    ('1100')    ('1101')    ('1110')    ('0000')    ('0010')    ('0110')    ('1000')    ('1111')    ('1111')
```

Fig. 3: Creation of Message blocks



```
%generate a knn codebook
function [Codebook]=CodebookGenerate(k,n,DD)
N=1:k;
% Degree distribution
Codebook=zeros(k,n);
for i=1:n
    index=randsample(k,DD(i));
    Codebook(index,i)=1;
end
Codebook
end
```

```
Codebook =

     0     0     0     1     0     0     1     0     0     1     0     0     0     0     0     1     0     1     0     0     1     1     0     1     1
     1     1     0     0     0     0     1     0     1     0     0     1     0     1     0     0     0     0     0     1     0     1     1     1     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0     0     0     0     0     1     0     0     1     0
     0     0     1     0     1     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     0     0     0     0     0     1     0     1     0     1     1     0     1     0     0     0     0     0     0     0     0
     0     1     0     0     0     1     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0     0     0     0     0     1     0     0     1
     0     0     0     0     0     0     0     0     1     0     1     0     0     0     0     0     0     0     0     0     0     1     0     0     0     0
     0     0     0     0     1     0     0     0     0     1     0     0     1     1     0     0     0     1     1     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     1     1     0     0     0     0     0     0     1     0     0     1     0     0     0
     1     1     0     1     0     1     0     0     0     0     0     0     0     0     1     1     0     0     1     0     1     1     1     0     0     0     0     1
```

Fig. 4: Generating Code book

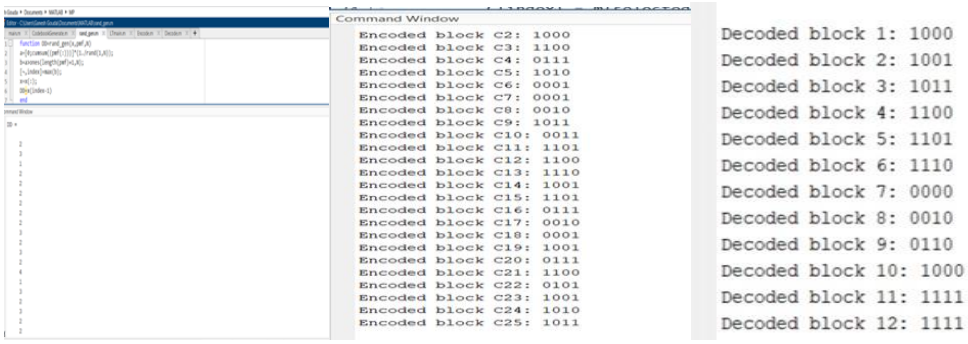


Fig. 5: Random number generation, Encoded, Decoded data

IV. IMPLEMENTATION ON FPGA

FPGA (Field-Programmable Gate Array) boards are highly versatile and powerful devices that offer numerous benefits across a wide range of applications. They allow users to create hardware designs by programming the underlying logic gates and interconnections. They provide excellent performance by allowing parallel processing capability and re-configurability. FPGA boards are valuable tools that combine the flexibility of software with the performance of hardware. Their customizable nature, high performance, and suitability for a wide range of applications make them an excellent choice for prototyping, accelerating computations, and implementing specialized hardware designs. In this work, LT codes encoding and decoding is implemented on DE10 Lite FPGA Board by writing the algorithms using Verilog code. The DE10-Lite presents a robust hardware design platform built around the Altera MAX 10 FPGA. The MAX 10 FPGA shown in fig(6) is well equipped to provide cost effective, single-chip solutions in control plane or data path applications and industry-leading programmable logic for ultimate design flexibility.

Verilog code is dumped into DE10 Lite FPGA board in Quartus Prime Integrated Development Environment (IDE) with modelsim simulation tool. The software will perform synthesis, place and route, and generate the programming file when Verilog code is compiled (fig(7). After configuring the FPGA, verification and testing is done for the program. Fig(8) shows the encoded data after running the Verilog code.

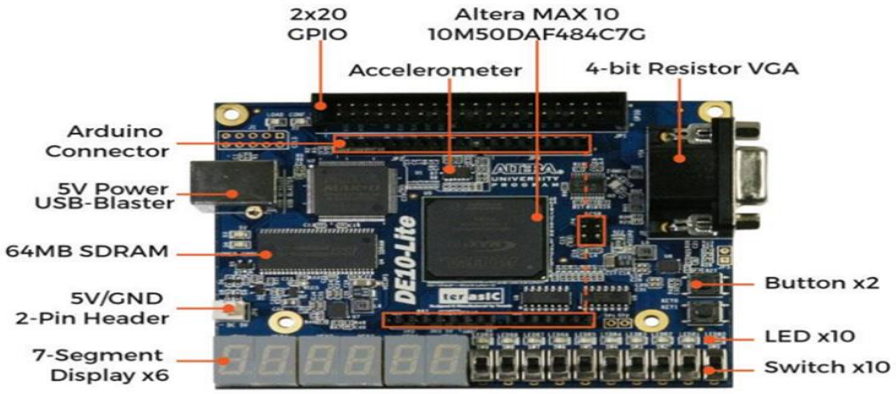


Fig. 6: DE10 Lite FPGA Board

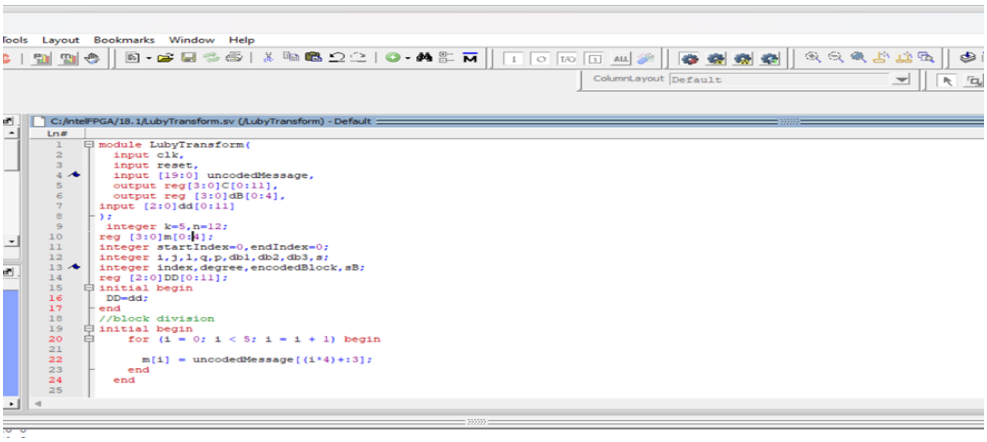


Fig. 7: Verilog code

ISSN No:-2456-2165

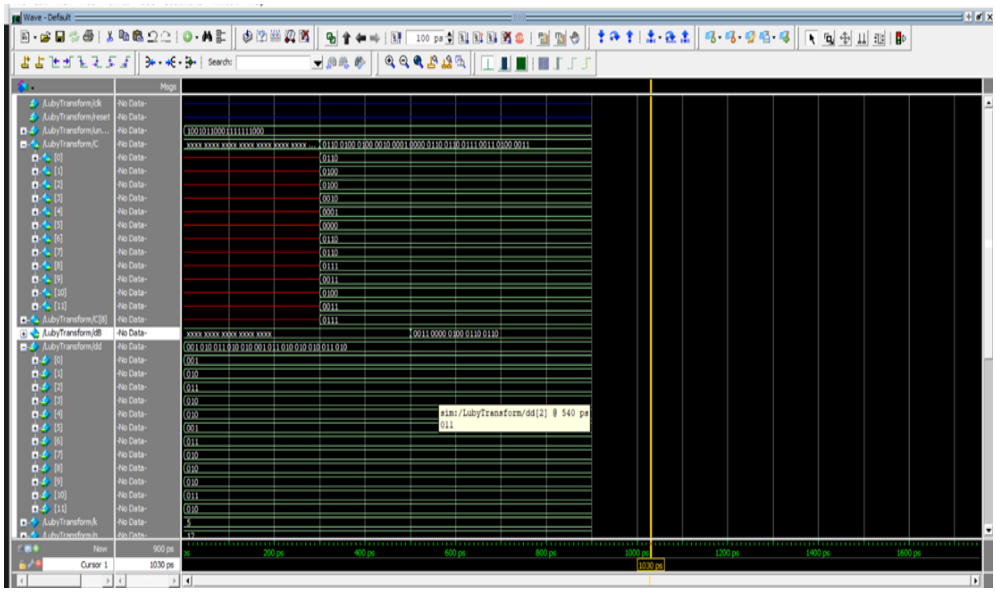


Fig. 8: Luby transform codes generation and decoding

The advantage of the LT approach is that it can provide high error correction capability using a small number of encoded packets. This is because the random linear codes used in the LT have a low-density structure, which makes them particularly effective at correcting errors in noisy communication channels. Overall, the Luby Transform is a powerful FEC technique that is widely used in many modern communication systems, including wireless networks, satellite communications, and digital television.

The ratelessness property of LT codes makes them attractive for applications where the channel conditions are uncertain or variable, such as wireless networks, satellite communications, or peer-to-peer file sharing systems. By generating an infinite number of coded packets, LT codes can provide flexible and adaptive error correction capabilities, without requiring complex feedback mechanisms or extensive pre-coding of data.

Though probabilistic decoding is commonly used to decode LT code, new decoding algorithms used in LDPC decoding such as bit flipping, message passing and belief propagation algorithms be used to decode LT codes when noise in the channel is more and varying. These algorithms further can improve the performance of the communication system.

V. CONCLUSION

The LT code generation and decoding algorithm is known for its simplicity and efficiency, as well as its ratelessness, which allows it to adapt to varying channel conditions. However, the trade-off is that it may require a larger number of encoding symbols to be transmitted than other error correction codes, which can impact the overall transmission efficiency. In this work, we have shown the generation and decoding using MATLAB. Later for hardware implementation, DE10 Lite FPGA board is used. Programming for this is written in Verilog. Performance analysis of code and hardware efficiency will be considered in the next work.

REFERENCES

[1.] M. Luby, “LT codes”, The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. proceedings.

[2.] David J. C. MacKay. 2002. “Information Theory, Inference & Learning Algorithms” Cambridge University Press, USA

[3.] R. Palanki, Jonathan S. Yedidia, “Rateless codes on noisy channels” International symposium on Information Theory, 2004. ISIT 2004 Proceedings. DOI: 10.1109/ISIT.2004.1365075

[4.] Iqbal Hussain, Ming Xiao, Lars K. Rasmussen Design of LT Codes with Equal and Unequal Erasure Protection over Binary Erasure Channels, 15 Jan 2013-IEEE Communications Letters

[5.] Anmol Singh Brar and Amandeep Singh Sandhu “Fountain Coded Wireless Transmission Model” Indian Journal of Science and Technology, Vol 9(14), DOI: 10.17485/ijst/2016/v9i14/86604, April 2016

[6.] Xin Song, Tuofeng Lei, Lingfeng Cheng “A Novel LT Encoding Algorithm with Low Error Floor” 2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP) DOI: 10.1109/ICSP51882.2021.9408964

[7.] Khaled F. Hayajneh, “Memory-Based LT Codes for Efficient 5G Networks and Beyond” Electronics 2021, 10, 3169. https://doi.org/10.3390/electronics10243169

[8.] S. Chanayai and A. Apavatjrut, “Fountain codes and their applications: Comparison and implementation for wireless applications,” Wireless Personal Communications, vol. 5, no. 12, pp. 1–16, 2021.

[9.] Guru Prasanth Sarvepalli; Sivaraman Rethinam etal “Secure Image Communication using Systematic-LT Codes over AWGN Channel” 2022 2nd Asian Conference on Innovation in Technology (ASIANCON) DOI: 10.1109/ASIANCON55314.2022.9909507

[10.] Joe Louis Paul Ignatius, Sasirekha Selvakumar “Enhanced Distributed Storage System Using Lower

Triangular Matrix-Luby Transform Codes" Intelligent Automation & Soft Computing 2022, vol.33, no.3 DOI:10.32604/iasc.2022.024173

[11.] K.L.Sudha , Om Prakash Singh , Pavan Mahalingappa Gunaki , Harshitha K , Sanju reddy "Packet Drop Rectification Using Luby Transform Code In A Wireless Communication Network-Implementation With Arduino" IOSR Journal of Engineering (IOSRJEN) www.iosrjen.org ISSN (e): 2250-3021, ISSN (p): 2278-8719 Vol. 09, Issue 7 (July. 2019), ||S (II) || PP 16-21