SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TNC TRANSACTIONS ON NETWORKS AND COMMUNICATIONS

VOLUME 4, ISSUE 5
ISSN: 2054 -7420

Performance Evaluation of Low Density Parity Check (LDPC) Codes over Gigabit Ethernet Protocol

Vinaya R. Gad, Udaysingh V. Rane, Rajendra S. Gad and Gourish M. Naik
Altera System on Chip Laboratory; Department of Electronics, Goa University Goa, India 403206;
rsgad@unigoa.ac.in

ABSTRACT

Error Correcting Low density Parity Check codes enable the communication systems to have a low-power, reliable transmission over noisy channels and can achieve data rates very close to Shannon limit when iteratively decoded. They are used in many digital communication systems such as digital video broadcasting (DVB-S2), MIMO-WLAN (802.11n), WMAN (802.16e), mobile broadband wireless access (MBW A) (802.20) and have a very good error correcting performance over a variety of channels. In this paper we present a performance platform for simulation studies of of LDPC decoding algorithms. We present the results of the simulation studies of Bit Error Rate (BER) performance for various block length like 64 and 256 bytes frame over Additive White Gaussian Noise (AWGN) channel. The comparative studies are made for Log Domain and Log Doman Simple decoding algorithms.

***Keywords**:* LDPC, BER, PER, FPGA, Gigabit Ethernet.

1 Introduction

There is a growing literature on the practical design of LDPC codes [1]. To give few examples of ¼ code rate like irregular LDPC code over GF(8), blocklength 48 000 bits (Davey, 1999); turbo code (JPL, 1996) blocklength 65 536; Regular LDPC over GF(16), blocklength 24 448 bits (Davey and MacKay, 1998); irregular binary LDPC code, blocklength 16 000 bits (Davey, 1999); Luby et al. (1998) irregular binary LDPC code, blocklength 64 000 bits; Turbo code for Galileo; Regular binary LDPC code: blocklength 40 000 bits (MacKay, 1999b); they are now being adopted for applications from hard drives to satellite communications and quantum error-correction (MacKay, 2004). Although the primary goal of any error correcting code is to achieve a performance that is close to the Shannon limit, one has to realized themselves to practical implementations for limited resources on the computing machine, latency of the operating systems, Complex Instructions Set Computing (CISC) instructions types on computing cores and processing time; all by limiting the iterations involved in complex algorithms so they can be integrated into real systems. Error correction algorithms are often implemented in hardware for fast processing to meet the real-time needs of communication systems.

DOI: 10.14738/tnc.45.2035
Publication Date: 02nd October 2016
URL: http://dx.doi.org/10.14738/tnc.45.2035

2 LDPC decoding algorithms: Sum-Product Algorithm – Probability Domain and Log Domain

The direct implementation of the decoding algorithm for binary codes in the probability domain (i.e., the SPA) has several drawbacks as compared to an implementation. SPA used in decoding of LDPC codes requires a large number of multiplications of probabilities which makes the algorithm numerically unstable, especially for very long codes. It may be noted that the best performance is obtained for very long codes. This long block length, combines with the need for iterative decoding, introduces latency which is unacceptable in many applications. Thus, a log-domain version of the algorithm is preferred, denoted here by log-SPA, based on log-likelihood ratios (LLR): the direct implementation is more sensitive to quantization effects and requires more quantization levels than when using LLRs [2,3].

We define the following log likelihood ratios as part of the decoding algorithmWe define the following log likelihood ratios as part of the decoding algorithm: $Lc_i = \log\left(\left(P_r(x_i = +1 \mid y_i)\right)/\left(P_r(x_i = -1 \mid y_i)\right)\right)$; $Lr_{ji} = \log(r_{ji}(0)/r_{ji}(1))$; $Lq_{ji} = \log(q_{ji}(0)/q_{ji}(1))$; $LQ_i = \log(Q_i(0)/Q_i(1))$. This algorithm iterates over columns and rows of parity check matrix and operates on nonzero entries by performing the following steps:

Step 0: Initialize Lq_{ji} by: $Lq_{ji} = Lc_i = 2y_i/\sigma^2$;Which initialized the values of coefficients $Q(x)_{ij}$ over logarithmic scale for the received symbols 'y_i' obtained after hard-decision decoder of the received decoded vector having 'σ' as standard deviation of the noise over the channel.

Step1: Evaluate Lr_{ji} by:

$$Lr_{ji} = \left(\prod_{i'\varepsilon R_{j\setminus i}} \alpha_{ji'}\right).\phi\left(\sum_{i'\varepsilon R_{j\setminus i}} \phi(\beta_{ji'})\right) \quad ; \tag{1}$$

where, $\alpha_{ji} = sign(Lq_{ji})$, $\beta_{ji} = \left\|Lq_{ji}\right\|$, $\phi(x) = -\log(\tanh(x/2)) = \log\left(e^x + 1/e^x - 1\right)$.

Step2:

$$Lq_{ji} = Lc_i + \sum_{j'\varepsilon C_i \setminus j} Lr_{j'i} \quad .$$

Step 3:

$$LQ_i = Lc_i + \sum_{j\varepsilon C_i} Lr_{ji} \quad . \tag{2}$$

Step 4: For every row index i:

$$\hat{c}_i = \begin{Bmatrix} 1 & if\ LQ_i < 0 \\ 0 & else \end{Bmatrix} . \tag{3}$$

If $\hat{c}H^T=0$, or if maximum number of iteration is reached then stop, else continue iterations from Step 1. The SPA requires message multiplications, whereas the log-SPA implementation uses message additions. The latter is more efficient in fixed point implementations, as fixed point multiplications can take up many clock cycles compared to additions.

2.1 Min-Sum Algorithm

Consider the update equation 1 for 'Lr_{ji}' in the log domain algorithm:

$$Lr_{ji} = \left(\prod_{i'\varepsilon R_{j\setminus i}} \alpha_{ji'}\right).\phi\left(\sum_{i'\varepsilon R_{j\setminus i}} \phi(\beta_{ji'})\right) \tag{4}$$

The 'ϕ(x)' is a function which is decreasing for the values of x > 0. It is intuitive that the term corresponding to the smallest β_{ji} in the above summation dominates, so that the second term in above equation: $\phi\left(\sum_{i'\varepsilon R_{j\setminus i}} \phi(\beta_{ji'})\right) = \phi(\phi(\min_{i'} \beta_{ji'})) = \min_{i'} \beta_{ji'}$. The second equality follows from ϕ (ϕ (x)) = x. Thus the Min-Sum algorithm is the same as SPA in which Step(1) is replaced by this equation: Step 1': $Lr_{ji} = (\prod_{i'\varepsilon R_{j\setminus i}} \alpha_{ji'}).(\min_{i'\varepsilon R_{j\setminus i}} \beta_{ji'})$. Because of the approximation in this equation, there is degradation in the performance of Min-Sum compared to SPA.

3 Simulation platform relevance in Optimizing ECC's

The Simulation platform comprises of the platform design for Gigabit Ethernet protocol, which is interfaced with the Matlab(Version 7.0) simulation model for BER performance analysis.

3.1 Platform Design

The platform for Gigabit Ethernet protocol is implemented using Altera's Triple Speed Ethernet (TSE) softcore IP Megacore function [4]. The design has been implemented on Altera's Stratix II GX PCI Express development board and the device used is 'EP2SGX90FF1508C3'. The said device is a Field Programmable Gate Array (FPGA) with '90960' Configurable Logic Blocks (CLBS), 717 MHz of maximum clock frequency, PBGA1508 package with Combinatorial Delay of a CLB-Max of 4.45 ns. The Fig. 1 shows the simplified block diagram of the Gigabit Ethernet protocol platform.

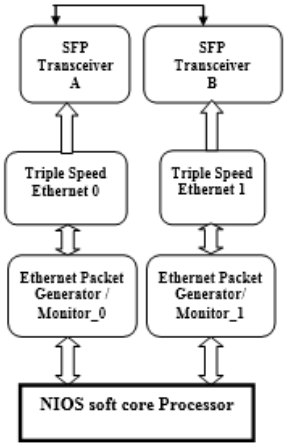

Figure. 1: Block Diagram of the Gigabit Ethernet protocol platform

The design includes two Altera TSE MegaCore functions, which implement the MAC, Physical Coding Sublayer (PCS) and Physical Media Attachment (PMA) sublayer in Full Duplex mode. There are two SFP (Small Form-factor pluggable) modules cages built onto the FPGA board, which provide the physical media interface. The Nios II processor is used to generate and monitor the Ethernet packets. The TSE MegaCore function handles the transmission and reception of the packets, which can be looped back using SFP modules with an Ethernet copper cable, fibre optic cable, or a switch using proper physical medias SFP. The design is built using Altera's Quartus II software and System On Programmable Chip (SOPC) builder. The design uses, on-chip memory of 256 Kbytes and the SOPC builder system uses a clock source of 83.33MHz. Altera's TSE design has been implemented and used as a platform for studying the performance of Gigabit Ethernet protocol Standards 1000Base-LX, 1000Base-SX and 1000Base-T.

There are three main components of the experimental setup as shown in Fig. 2(a). I. Stratix II GX PCI Express development board having device 'EP2SGX90FF1508C3': This is the platform for the Gigabit Ethernet protocol design wherein the algorithms are synthesized. II. SFP transceivers cages on the Stratix II GX FPGA: For interfacing different types of physical media of Ethernet like Cat5e, Cat6, Single mode fibre and Multi-mode fibre using SFP Transceivers. III. Fibre mounting and positioning three axis linear stages stand: Platform for mounting the fibre and introducing calculated optical attenuation in channel by lateral, longitudinal displacements of fibre joints. Now errors are introduced in the fibre channel by longitudinal displacement using the adjustment screw along the x-axis of the fibre mounting and positioning stand. The graph in Fig. 2(b) illustrates the packet loss introduced in the fibre with longitudinal displacement, which is equivalent to AWGN channel in the experimental setup. This confirms the Ethernet frame generator system development on configurable devices ready to be used for ECC platforms explain in next section. Figure 3 shows a MATLAB model developed to study the BER performance of Gigabit Ethernet protocol using LDPC codes. This model can be used for error correction performance analysis of Gigabit Ethernet protocol.

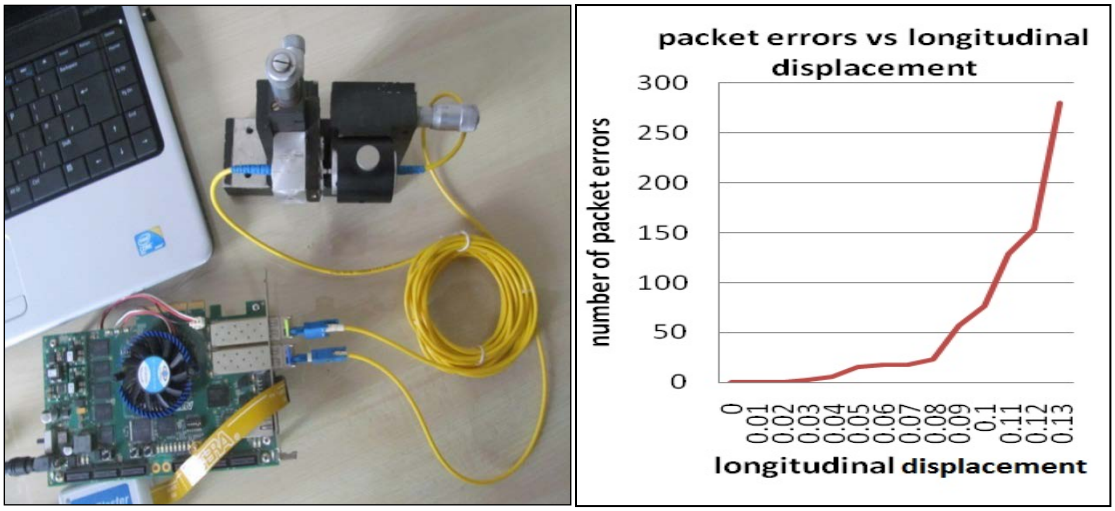

Figure 2: (a) Experimental Setup; (b): Packet errors vs longitudinal displacement.

The Ethernet frame is generated using ALTERA TSE IP core generator as shown in Fig. 1 and these frames are captured using WireShark and given to the Matlab simulation model of error correction shown in Fig. 3. The frames are encoded using the generator matrix. The BPSK modulator scheme maps the input binary signals, to an analog signal for transmission. The AWGN channel is the medium through which information is transmitted from the transmitter to the receiver for introducing errors in transmitted frames. The AWGN channel is representing the noisy physical media like a copper, fibre optic or wireless channel. The LDPC decoder is implemented at the receiver. Here, two decoding algorithms used i.e. Sum Product Algorithm (SPA)-logdomain and SPA-Min-Sum Algorithm, which loops through passing messages back and forth along the tanner graph until maximum number of passes have occurred. The estimated message is compared to the transmitted message at the receiver end in order to detect whether there was an error in transmission.

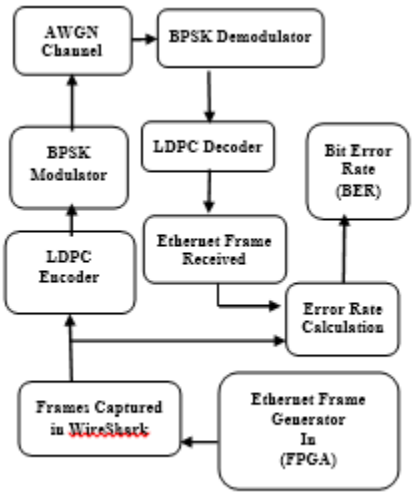

Figure 3: Simulation model for Error Correction interfaced with Triple Speed Ethernet design.

4 Results and Discussions

The LDPC decoding algorithms which are efficient for hardware implementation i.e. Sum Product Algorithm (SPA)-logdomain and SPA-Min Sum Algorithm (logdomainSimple) are chosen for simulation studies. The IBM X3200 server having one Intel quad-core (Xeon 3400 series) with 32 KB instruction cache, 32 KB data cache, and up to 8 MB L3 cache shared among the cores and support for Intel Extended Memory 64 Technology (EM64T) with 1333 MHz 32GB SDRAM DIMM has been set as a computing machine for model computation. The performance studies (Figures 4 and 5) of LDPC codes were computed for 100 numbers of Ethernet Frames having block length of 512 and 1024 bits corresponding to 64 and 256 bytes for number of iterations from 5 to 50 frame lengths respectively. The simulations were limited to lower value of frame lengths over higher iteration due to linear computation time. It may be noted that computation time for the block length of 512 (i.e. Figure 5.1. (a)) over 5-20 iterations was approximately 40 hours.

Results, as illustrated in Figure 4 to 5 indicate that the BER performance improves with increase in block length which can be optimized for error correction by setting the number of decoding iterations between 5 to 50 for a SNR close to 2dB. Since, more errors are introduced for larger frame lengths and hence the number of iterations needs to be increased for better Error Correction performance. It may be noted that for higher number of iterations 20 to 300 for block length of 512 bits corresponding to 64 bytes frame lengths there is no improvement in the error corrections, while obtaining performance of close to 10^{-3} BER at 2 SNR.

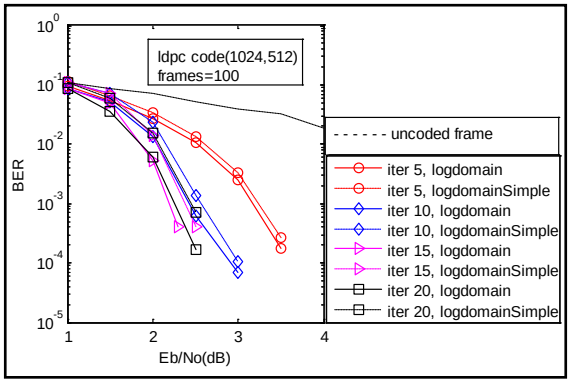

Figure. 4: BER vs SNR for 64 bytes frame over 5 – 20 iterations

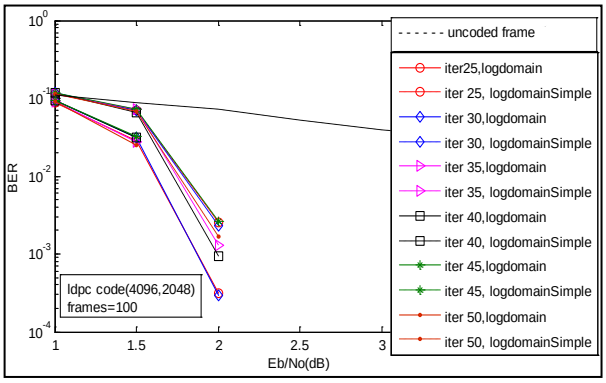

Figure. 5: BER vs SNR for 256 bytes frame over 25 – 50 iterations

There are growing literatures on the practical design of LDPC codes; they are now being adopted for applications from hard drive to satellite communication. Khaled Shuaib et al, focuses on developing MatLab/Simulink models for the Zigbee protocol and the performance evaluation of these models. Several simulations were carried and the results were analyzed for the different scenarios. The results show how the relationship between the signal Bit Error Rate (BER) and Signal to Noise Ratio (SNR) was affected when varying the data rate and power [5]. David J. C. MacKay gives experimental results for binary-symmetric channels and Gaussian channels demonstrating the practical performance of LDPC codes is substantially better than that of standard convolutional and concatenated codes and is almost as close to the Shannon limit as that of turbo codes[6]. Murad Hossain et al presents a study of the errors observed when an optical Gigabit Ethernet link is subject to attenuation. They have introduced modified log-domain algorithm and min-sum algorithm of sum-product algorithm (SPA) for LDPC codes over GF (q) and compared the BER performance and computational complexity of the log domain algorithm, modified log-domain algorithm and modified min-sum algorithm. The BER performance of modified log domain decoding and modified min-sum decoding is better than log domain decoding algorithms [7]. Ganepola V. S. et al establish working in a higher order Galois field, significantly improve the performance of the LDPC code with moderate code lengths[8]. Hua Xiao et al proposed method for estimating the performance of low-density parity-check (LDPC) codes decoded by hard-decision iterative decoding algorithms on binary symmetric channels (BSCs) is proposed [8]. Chris Howland et al designed a 1024 bit rate 1/2 LDPC code decoder and implemented which matches the coding gains of equivalent turbo codes. This parallel decoder architecture supports throughputs upto 1 Gbps[9].

REFERENCES

[1] Yongyi Mao, Amir H. Banihashemi, Decoding Low-Density Parity-Check Codes With Probabilistic Scheduling, IEEE Communications Letters, 5(10)(2001).

[2] Bernhard, M. J. Leiner, LDPC Codes – a brief Tutorial , 2005.

[3] L. Ping , W. K. Leung, 'Decoding low density parity check codes with finite quantization bits'; IEEE Comm. Letters, 4(2)(2000)62–64.

[4] S. Khatri, R. Brayton, A. Sangiovanni-Vincentelli, Cross-talk Immune VLSI Design Using a Network of PLAs Embedded in a Regular Layout Fabric', IEEE/ACM International Conference on Computer-aided Design, (2000) 412–418.

[5] K. Shuaib, M. Alnuaimi, M. Boulmalf, I. Jawhar, F. Sallabi, A. Lakas, Performance evaluation of IEEE 802. 15. 4: experimental and simulation results, Journal of Communications, 2(4)(2007) 29–37.

[6] D. MacKay, Good error correcting codes based on very sparse matrices, IEEE Trans. Information Theory,(1999)399-431.

[7] Md. Murad Hossain, Md. Rakibul Islam, Md.Jahidul Islam, Asif Ahmed, Md. Shakir Khan, S. M. Ferdous, Modified Log Domain Decoding Algorithm for LDPC Codes over GF(q), Journal of Selected Areas in Telecommunications (JSAT), (2011).

[8] Hua Xiao, Amir H. Banihashemi, Estimation of Bit and Frame Error Rates of Finite-Length Low-Density Parity-Check Codes on Binary Symmetric Channels, IEEE Transactions On Communications, 55(12)(2007).

[9] Chris Howland, Andrew Blanksby, A 220 mW 1Gb/s 1024-Bit Rate-1/2 Low Density Parity Check Code Decoder, IEEE 2001 Custom Integrated Circuit Conference;(2001), 0.7803-6591-7/01.