TELKOMNIKA, Vol.11, No.11, November 2013, pp. 6999~7006
e-ISSN: 2087-278X

A Frame Synchronization based on Likelihood-Ratio Threshold of LDPC Decoding

Zhixiong Chen*, Dongsheng Han, Hongyin Xiang
Department of Electronic and Communication Engineering, North China Electric Power University, Baoding, China, 071003
*Corresponding author, e-mail: chenzx1983@sohu.com

Abstract

A LDPC code-aided frame synchronization algorithm based on likelihood ratio value threshold is proposed and analyzed in this paper. For soft information vectors with different frame offsets, the likelihood ratio value presenting the probability ratio of satisfaction of all check restrictions and dissatisfaction of all check restrictions are computed respectively, and then the point larger than a threshold is chosen as the final frame synchronization bounder. The method to select the threshold is studied theoretically and is analyzed according to simulation result. And Frame synchronization error ratio of new algorithm is derived. Compared to exiting LDPC code-aided frame synchronization algorithm based on maximum LLR, new algorithm saves synchronization searching time. Synchronization performance simulation results of new algorithm are given and analyzed.

Keywords: *frame synchronization, maximum-Likelihood rule, LDPC code-aided, threshold*

1. Introduction

Because of excellent decoding performance, LDPC codes [1] have been adopted in several standards: IEEE 802.11n and IEEE 802.16e, etc. However, the frame synchronization ability for communication receivers should be improved and satisfied firstly in order to realize the full potential of LDPC codes at low SNR where symbols are badly corrupted by noise and disturbance.

Since conventional frame synchronizers require significant portion of the bandwidth and signal energy at low SNR to ensure the synchronization performance. The LDPC code-aided frame synchronization algorithms using no pilot symbols attract great attention. Matsumoto and Imai [2] observed that frame synchronization failure reduces the amplitude mean of the extrinsic log-likelihood ratios (LLR) of all variable nodes. Another synchronization algorithm which correlates the received symbols against the LDPC decoded soft symbols is proposed in [3]. In order to avoid the local maximum convergence problem of the EM algorithm, the authors propose a discrete EM method which locates the global maximum via an exhaustive search. These algorithms have the downside of the need for a full LDPC decoding iteration and additional hardware resource to implement the algorithm.

The hard decision frame synchronization [4-5] proposed by Dong-U Lee et el. computes the ratio of satisfied constraint node equations with different frame offsets to estimate the true frame boundary. The rationale is that the number and nature of satisfied constraint node equations provide a measure of underlying accuracy of the frame offset estimates. Realized by simple XOR operations as main advantage, the algorithm causes unacceptable performance degradation because of the information loss by hard decision. At least two consecutive frames are needed to estimate frame offset to get acceptable frame synchronization performance of the code-aided algorithm based on soft information [5]. A novel Code-aided frame synchronization algorithm using Quasi-Cyclic LDPC soft decoding information based on Maximum-Likelihood rule is investigated and simulated in [6]. The frame synchronization algorithm can be implemented by parts of resource of LDPC decoder and get favorable performance than the hard decision frame synchronizations.

Continue the research above, another code-aided frame synchronization algorithm named as LLR threshold method based algorithm is studied and analyzed in depth in this paper. Main contribution of this paper is the analysis of probability density function (PDF) for LDPC

Received May 27, 2013; Revised August 5, 2013; Accepted August 16, 2013

decoding LLR value, the derivation of frame synchronization error ratio (FSER) of new algorithm and the simulation for the PDF and FSER. The outline of the paper is as follows. In Section II, LDPC decoding algorithm is provided and discussed. Section III describes the proposed pilotless frame synchronization algorithms. Section IV derives the FSER of the code-aided frame synchronization algorithms. The performance of new algorithms is investigated by simulation in Section V and concluding remarks are given in Section VI.

2. Belief Propagation Decoding

In this section, it is explained the results of research and at the same time is given the comprehensive discussion. Results can be presented in figures, graphs, tables and others that make the reader understand easily [2], [5]. The discussion can be made in several sub-chapters.

It is assumed that the LDPC encoding frame is BPSK modulated and perturbed by an AWGN. The BP decoding algorithm can be summarized as follows:

1) Initialization: For every node couple (*i*, *j*) that $H_{i,j}=1$, the *ith* variable node is assigned an a posteriori value.

$$L(q_{ij}) = L(P_i) = 4y_i / N_0 \tag{1}$$

2) Check node processing: Denote α_{ij} and β_{ij} as the symbol and absolute value of $L(q_{ij})$ respectively. For the *jth* check node and the set $B(j)$, the process is done as:

$$L(r_{ji}) = \prod_{i' \in B(j)/i} \alpha_{i'j} \times \phi\left(\sum_{i' \in B(j)/i} \phi\left(\beta_{i'j}\right) \right) \tag{2}$$

Where the function $\phi(z)$ (z >0) is defined as:

$$\phi(x) = \phi^{-1}(x) = \log \frac{\exp(x)+1}{\exp(x)-1} \tag{3}$$

In order to reduce the hardware realizing complexity of the decoder, the expression (2) is calculated with two steps [7-8]. The first step computes the total symbol and absolute value for *jth* check node:

$$\eta_j = \prod_{i \in B(j)} \alpha_{ij} \tag{4}$$

$$\lambda_j = \sum_{i \in B(j)} \phi\left(\beta_{ij}\right) \tag{5}$$

Secondly the decoder computes the information send to the *ith* variable node.

$$L(r_{ji}) = \left(\alpha_{ij} \cdot \eta_j\right) \phi\left(\lambda_j - \phi\left(\beta_{ij}\right)\right) \tag{6}$$

3) Variable node processing: For each variable node, computations are done as:

$$L(q_{ij}) = L(P_i) + \sum_{j' \in C(i) \setminus j} L(r_{j'i}) \tag{7}$$

$$L(Q_i) = L(P_i) + \sum_{j \in C(i)} L(r_{ji}) \tag{8}$$

4) Decision: Quantize C' such that c_i=0 if P_i≥0, and c_i=1 if P_i<0. If $C'H^T$=0, halt, with as the decoder output; otherwise go to step 2).

3. LDPC Aided Frame Synchronization Algorithm based on LLR Threshold

It is assumed that the LDPC encoded frames are BPSK modulated and perturbed by an AWGN channel with zero mean and variance of N0/2 where N0 is the one-sided noise spectral density. The system model is shown in Figure 1.

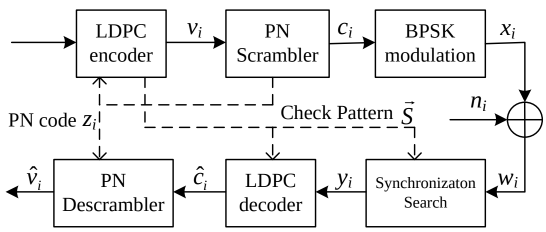


Figure 1. QC-LDPC Coding System Model

The received signal is buffered in the synchronizer as shown in Figure 2, where *N* is the frame size, *m* is the true frame boundary, and *T* is the number of consecutive frames used for synchronization. The task of the frame synchronization algorithm is to find an estimate *u* of *m* where $u \in py=[m-L, m+L]$. If $u = m$, then a frame synchronization is achieved.

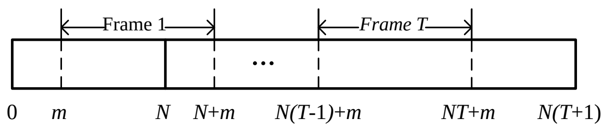


Figure 2. Synchronization Buffer Structure

Based on ML rule, a code-aided frame synchronization algorithm based maximum LLR value is researched in [6]. The algorithm computes a likelihood ratio of probability $LLR(\vec{W}(u))$ between the probability of $\Pr[\vec{W}(u)]$ and $\overline{\Pr}[\vec{W}(u)]$with frame offset $u \in (m-L, m+L)$. And a new synchronizer based on the maximum LLR values estimates the frame offset u as follows:

$$\hat{u} = \arg\max_{u \in py} \sum_{j=1}^{M} (1-2s_j)\left(\eta_j^{(u)} \phi\left(\lambda_j^{(u)}\right)\right) \tag{9}$$

Where $\vec{S} = [s_1, s_2, ..., s_M]$ is the check result of vector $\vec{C}$ multiplied by the parity-check matrix $H_{M \times N}$, computed in a way as $\vec{S} = \vec{C}(H)^T = (\vec{V} \oplus \vec{Z})(H)^T = \vec{Z}(H)^T$.

Because of the usage of PN sequence, the BP decoding algorithm should be modified accordingly. As mentioned, BP decoding algorithm computes the pseudo-posterior *LLRs.*

$$P_n = \log \frac{\Pr\{c_n = 0 | y_n\}}{\Pr\{c_n = 1 | y_n\}} \tag{10}$$

Note that the purpose of check node processing is to compute the *LLR* values sent for all the variable nodes that participate in the check node *j.* The *LLR* value $L(r_{ji})$ from check node *i* presents the logarithmic probability ratio of the variable node *i* being zero and one with the other variable nodes' LLR knowledge computed and sent from the last iteration. Without any change of the other decoding steps, only the check node processing is modified as follows:

$$L(r_{ji}) = (1-2s_j) \times \prod_{i' \in B(j)/i} \alpha_{i'j} \times \varphi\left(\sum_{i' \in B(j)/i} \varphi\left(\beta_{i'j}\right)\right) \tag{11}$$

Where $s_j = 0$ presents that there is even number of 1 in the *j*th check equation and the processing is the same as (2); $s_j = 1$ presents odd number of 1 in the *j*th check equation and the processing should be modified by multiplying (2) with -1 to convert the probability ratio of *i*th variable node.

Based on the $LLR(\vec{W}(u))$ computed in the maximum LLR value algorithm of (11), the other LDPC code-aided frame synchronization algorithm based on the threshold method with a threshold of τ is proposed accordingly as follows:

$$\hat{u} = \arg\min_{u \in W} u \tag{12}$$

Where,

$$W = \left\{ u \left| \sum_{j=1}^{M} (1 - 2s_j)\left(\eta_j^{(u)}\phi\left(\lambda_j^{(u)}\right)\right) > \tau, u \in py \right. \right\} \tag{13}$$

Obviously, the frame synchronization algorithm based on LLR threshold method dose not need a total iterative decoding procession and is realized by parts of hardware resource of decoder. According to (12) and (13), we do not need to compute all the LLR value for *N* offsets. Begin with $u = 0$, when the first LLR value is computed larger than τ, the frame synchronization searching is terminated and we get the frame boundary. The algorithm raises the speed of synchronization compared to the algorithm based on maximum method in [6].

4. Derivation of Frame Synchronization Error Ratio

A soft threshold value τ is set in the new frame synchronization algorithm proposed, and than the offset according to the first computing $LLR[\vec{W}(u)]$ that is larger than the threshold is estimated as the frame boundary. So the define of threshold τ is important for the performance of new algorithm. It is necessary to analyze the PDF of $LLR[\vec{W}(u)]$ in both synchronizing and unsynchronizing situation to direct the choice of τ.

If and only if the computing $LLR[\vec{W}(u)]$ with true frame offset $\bar{u}$ is larger than the threshold and others $LLR[\vec{W}(u)]$ with offset $u \in [0, \bar{u} - 1]$ is smaller than τ, a correct frame boundary estimation by new algorithm of threshold method is done. Suppose the true offset u is a uniform distribution variable in the range of [0, *N*-1] similar to [5].

Denote *f*(*t*) as the probability density function (PDF) of $LLR(\vec{W}(u))$ under frame synchronization with the offset of $\bar{u}$, and $g_i(t)$ as the PDF of $LLR(\vec{W}(u))$ with the offset of *i* ($i \in [0, N-1]$, $i \neq \bar{u}$).

Denote FSR_{mx} as the correct frame synchronization estimation probability of new proposing algorithm based on LLR threshold method and $FSER_{mx}$ as the wrong frame synchronization estimation probability. Then FSR_{mx} of blind frame synchronization based on LLR threshold is provided as follows:

$$FSR_{mx} = \frac{1}{N}\sum_{u=0}^{N-1}\int_{\tau}^{+\infty} f_u(t)\mathrm{d}t\left(\prod_{i=-1}^{u-1}\int_{-\infty}^{\tau} g_i(t)\mathrm{d}t\right) \tag{14}$$

Where $f_u(t)$ ($u \in [0, N-1]$) is the PDF of $LLR(\vec{W}(u))$ in the synchronization case. Because the synchronization value $LLR(\vec{W}(u))$ is relative to the channel SNR and check matrix, $f_u(t)$ has nothing to do with the value of u, that is:

$$f_0(t) = f_1(t) = \cdots f_{N-1}(t) = f(t) \tag{15}$$

Denote $g_i(t)$ as the PDF of $LLR(\vec{W}(i))$ with the offset of *i* ($i \in [0, N-1]$, $i \neq u$) under synchronization situation. Under the scramble of PN sequence, for the same LDPC code, the computing $LLR(\vec{W}(u))$ is unequal with different offset. As a result the $g_u(t)$ is different with different offset *u*. In order to simplify the expression, $g_{-1}(t)$ is imported and satisfies $\int_{-\infty}^{\tau} g_{-1}(t)\mathrm{d}t = 1$.

Then the $FSER_{mx}$ is computed according to the following formula:

$$FSER_{mx} = 1 - \frac{1}{N}\int_{\tau}^{+\infty} f(t)\mathrm{d}t \sum_{u=0}^{N-1}\left(\prod_{i=-1}^{u-1}\int_{-\infty}^{\tau} g_i(t)\mathrm{d}t\right) \tag{16}$$

From (16) the value of $FSER_{mx}$ is relative to the threshold τ, $f(t)$ and $g_i(t)$ directly. For the same SNR, $f(t)$ and $g_i(t)$ of $LLR[\vec{W}(u)]$ under both synchronization and unsynchronization situation is fixed, so the choice of threshold is a key factor to the frame synchronization performance based on LLR threshold method.

In order to compare the frame synchronization performance based on maximum method proposed in [6], $FSER_{mx}$ of the algorithm which has never been discussed is also derivate meanwhile in this section. The true offset u is also supposed to be a uniform distribution variable in the range of [0, *N*-1].

When the true system frame synchronization takes place in the offset $\bar{u}$ ($\bar{u} \in [0, N-1]$) and the computing $LLR[\vec{W}(u)]$ is equal to *x*, the probability of correct frame boundary estimation for the based on maximum method is:

$$\prod_{i\in[0,N-1],i\neq u}\int_{-\infty}^{x} g_i(t)\mathrm{d}t \tag{17}$$

Average the value *x*, we get the average probability of correct frame synchronization estimation:

$$\int_{-\infty}^{+\infty}\left(\prod_{i\in[0,N-1],i\neq u}\int_{-\infty}^{x} g_i(t)\mathrm{d}t\right) f(x)\mathrm{d}x \tag{18}$$

Because the true frame offset $\bar{u}$ satisfies the uniform distribution in the range of [0, *N*-1], if we average the value of (18), then the FSR_{mx} is provided.

$$FSR_{\max} = \sum_{u=0}^{N-1}\frac{1}{N}\int_{-\infty}^{+\infty}\left(\prod_{i\in[0,N-1],i\neq u}\int_{-\infty}^{x} g_i(t)\mathrm{d}t\right) f(x)\mathrm{d}x \tag{19}$$

And finally, $FSER_{mx}$ of the frame synchronization algorithm based on maximum method in [6] is acquired.

$$FSER_{\max} = 1 - \frac{1}{N}\sum_{u=0}^{N-1}\int_{-\infty}^{+\infty}\left(\prod_{i\in[0,N-1],i\neq u}\int_{-\infty}^{x} g_i(t)\mathrm{d}t\right) f(x)\mathrm{d}x \tag{20}$$

As the $g_u(t)$ of $LLR[\vec{W}(u)]$ with different offsets is different, it is difficult to compute the $FSER_{mx}$ of the frame synchronization performance based on LLR threshold and choose the optimal threshold. A representative PDF named as $g_*(t)$ with ordinary statistical parameter can be chose to represent other $g_u(t)$ in order to reduce the computing complexity. So we can define a identical mean and variance for $g_u(t)$, and the $FSER_{mx}$ of the proposed algorithm based on LLR threshold can be simplified as follows:

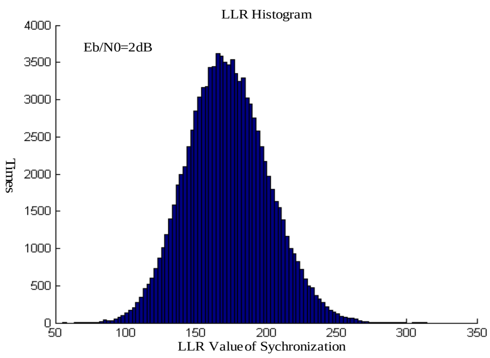
$$FSER'_{mx} = 1 - \frac{\int_{\tau}^{+\infty} f(t)\mathrm{d}t \times \left[1 - \left(\int_{-\infty}^{\tau} g_*(t)\mathrm{d}t\right)^N\right]}{N \times \left[1 - \left(\int_{-\infty}^{\tau} g_*(t)\mathrm{d}t\right)\right]} \quad (21)$$

Although a quasi-optimal threshold can be acquired by finding the derivative of $FSER_{mx}$ in (21) to the value τ, the need for computation dimension and value precision is great because of derivative to integral. A numerical simulation method can be used to define the quasi-optimal threshold, where $FSER_{mx}$ simulation with different SNR and threshold value is done and the threshold relative to the minimum $FSER_{mx}$ is chosen.

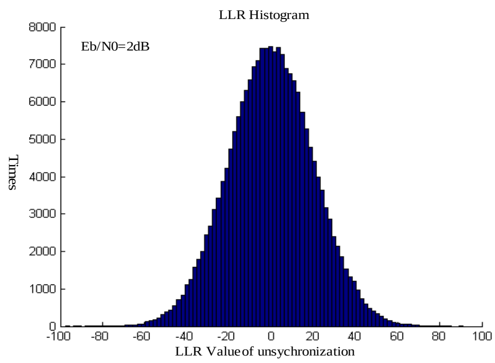
5. Simulation and Discussion

The (1944, 972) quasi-cyclic irregular LDPC code proposed for the IEEE 802.11n standard [9] is used in the simulation. The code determined by a parity-check matrix $H_{M\times N}$ has 810 degree-7 and 162 degree-8 constraint nodes. The quasi-cyclic LDPC code aided frame synchronization algorithm using PN sequence is summarized in (9).

The Figure 3 presents the histograms for $LLR[\vec{W}(u)]$ in the synchronizing and unsynchronizing cases when the E_b/N_0=2dB. Obviously, the value of $LLR[\vec{W}(u)]$ satisfy Gauss distribution. This conclusion is important to analyze and compute the theoretical FSER of the synchronization algorithm based on a threshold method of (12).



(a) frame synchronization situation



(b) frame unsychronization situation

Figure 3. Histogram for LLR Value in Synchronizing and Unsynchronizing Situation

The relation between the mean and variance of computing $LLR[\vec{W}(u)]$ in synchronizing and unsynchronizing situation and different E_b/N_0 is shown in figure 4. The mean of $g_u(t)$ is nearly equal to 0 and the variance of *f(t)* and $g_u(t)$ increase with the raise of E_b/N_0, which means the discretization of $LLR[\vec{W}(u)]$ increase with the improvement of channel condition. For different channel, the choice of the threshold is dynamic because of the difference of statistical parameters for $LLR[\vec{W}(u)]$.

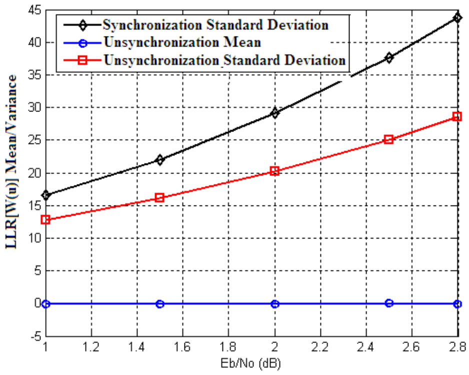


Figure 4. Mean and Variance for LLR Value with Different SNR

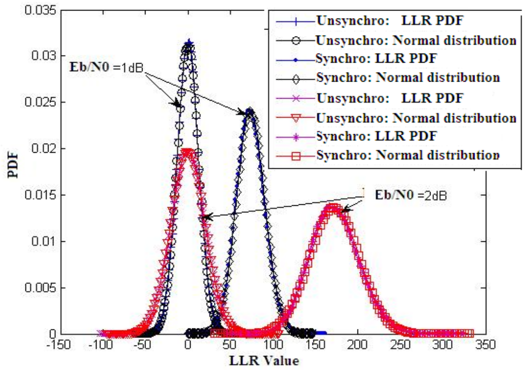


Figure 5. Comparison of the PDF of LLR in Synchronizing and Unsynchronizing Situation

The PDF line for $LLR[\vec{W}(u)]$ by computer simulation in both synchronizing and unsynchronizing situation is shown and compared to standard Gaussian distribution in Figure 5 when E_b/N_0=1dB and E_b/N_0=2dB. From the figure, it is obvious that simulation PDF line is almost consistent to theoretical line with the same mean and variance, satisfying the conclusion made in section above.

In Figure 5, when the SNR increases in synchronizing situation, the PDF line of $LLR[\vec{W}(u)]$ become smoother and get closer to the right side of figure. It means an obvious increase of mean and variance. For the unsynchronizing case, the mean of $LLR[\vec{W}(u)]$ is near to 0 and the variance increase with the raise of E_b/N_0.

There is an obvious overlapping part between the PDF lines of $LLR[\vec{W}(u)]$ in synchronizing and unsynchronizing situation, which can explain the occurrence of wrong frame synchronization decision for frame synchronization algorithm based on LLR threshold method. With the increase of SNR, distance of two PDF line become larger and the overlapping part get smaller, it is easier to distinguish the synchronizing and unsynchronizing situation and improve the performance of blind frame synchronizing algorithm.

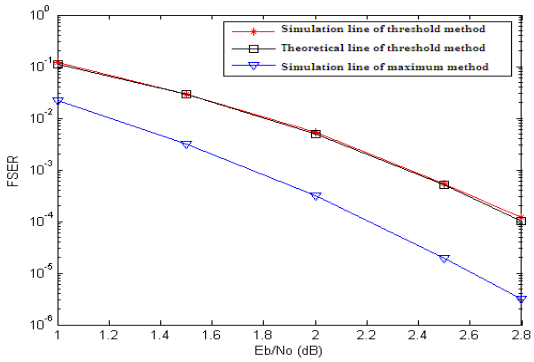


Figure 6. FSER of New Frame Synchronization Algorithms

In Figure 6, the FSER performance of new frame synchronization algorithms are given by simulation and compared. The theoretical FSER performance is almost the same as the simulation one. Because the searching range of the frame synchronization algorithm with the threshold method is not always equal to the frame length, its FSER performance lags behind the algorithm with the maximum LLR method.

6. Conclusion

In this paper, two code-aided frame synchronization algorithms based on the maximum LLR method and the threshold method respectively for Quasi-Cyclic LDPC coded system are investigated. And the FSER of the algorithms are given. The frame synchronization algorithms can be implemented by parts of resource of LDPC decoder saving the hardware resource to implement. Compared to hard decision algorithms, simulation shows good synchronization performance of the new algorithms.

Acknowledgement

The research was supported by “the Fundamental Research Funds for the Central Universities” under grant number 13MS66 and 13MS60.

References

[1] MacKay DJC. Good error correcting codes based on very sparse matrices. *IEEE Transaction on IT*. 1999; 45(2): 399-431.
[2] Matsumoto W, Imai H. *Blind synchronization with enhanced sum-product algorithm for low-density parity-check codes.* The 5th International Symposium on Wireless Personal Multimedia Communications, Hawaii. 2002; 966-971.
[3] Wymeersch H, Moeneclaey. ML frame synchronization for turbo and LDPC codes, Proceeding International Symp. *DSP and Communications Systems.* Australia. 2003; 15-22.
[4] Lee D, Kim H, Jones C, Vilasenor J. Pilotless frame synchronization via LDPC code constraint feedback. *IEEE Commun. Letter*. 2007; 11(8): 683-685.
[5] Lee D, Kim H, Jones C, Vilasenor J. Pilotless frame synchronization for LDPC-coded transmission systems. *IEEE Transactions on Signal Processing*. 2008; 56(7): 2865-2874.
[6] Chen ZX. LDPC Code-Aided Frame Synchronization Algorithms Based on LLR Value from Decoder. *International Review on Computer and Software*. 2012; 7(3): 1375-1379.
[7] Liu LZ, Richard Shi CJ. Sliced Message Passing: High Throughput Overlapped Decoding of High-Rate Low-Density Parity-Check Codes. *IEEE Transactions on Circuits and Systems*. 2008; 55(11): 3697-3710.
[8] Lee JY, Ryu HJ. A 1-Gb/s Flexible LDPC Decoder Supporting Multiple Code Rates and Block Lengths. *IEEE Transactions on Consumer Electronics*. 2008; 54(2): 417-424.
[9] IEEE P802.11nTM/D1.02, Draft Amendment to STANDARD Information Technology Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Enhancements for Higher Throughput, IEEE 802. 2006.