802.11n LDPC code

Simone Gaiarin

Course of Channel Coding and Capacity, AY 2011/2012

Table of contents

802.11n Standard

- In 2009 the IEEE accepted the 802.11n amendment, which improve the 802.11-2007 wireless networking standard allowing to obtain raw throughput up to 600 Mbit/s. The amendment introduce many improvements, i.e. MIMO techniques, 40MHz channels, and many others. Among these features, the standard adopts also the LDPC codes as an optional channel coding scheme

LDPC Codes

- LDPC codes are a class of error correcting code that can achieve Bit Error Rate (BER) close to the Shannon's limit.
- The LDPC codes are parity check codes which parity check matrix is sparse
  - The requirement on the sparseness of the matrix is necessary to guarantee that the decoder complexity remains low
  - This feature allow to use the message passing decoding technique

802.11n Parity-Check Matrix I

Staircase structure

0	-	-	-	0	0	-	-	0	-	-	0	1	0	-	-	-	-	-	-	-	-	-	-
22	0	-	-	17	-	0	0	12	-	-	-	-	0	0	-	-	-	-	-	-	-	-	-
6	-	0	-	10	-	-	-	24	-	0	-	-	-	0	0	-	-	-	-	-	-	-	-
2	-	-	0	20	-	-	-	25	0	-	-	-	-	-	0	0	-	-	-	-	-	-	-
23	-	-	-	3	-	-	-	0	-	9	11	-	-	-	-	0	0	-	-	-	-	-	-
24	-	23	1	17	-	3	-	10	-	-	-	-	-	-	-	-	0	0	-	-	-	-	-
25	-	-	-	8	-	-	-	7	18	-	-	0	-	-	-	-	-	0	0	-	-	-	-
13	24	-	-	0	-	8	-	6	-	-	-	-	-	-	-	-	-	-	0	0	-	-	-
7	20	-	16	22	10	-	-	23	-	-	-	-	-	-	-	-	-	-	-	0	0	-	-
11	-	-	-	19	-	-	-	13	-	3	17	-	-	-	-	-	-	-	-	-	0	0	-
25	-	8	-	23	18	-	14	9	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0
3	-	-	-	16	-	-	2	25	5	-	-	1	-	-	-	-	-	-	-	-	-	-	0

Systematic length

First W parity bits

802.11n Parity-Check Matrix II

- The 802.11n parity-check matrix has a particular staircase structure that allow to compute the parity bits using the back-substitution technique
  - The parity-check matrix is partitioned into square subblocks (submatrices) of size W x W
  - These submatrices are either cyclic-permutations of the identity matrix or null submatrices
  - The cyclic-permutation matrix P_i is obtained from the W x W identity matrix by cyclically shifting the columns to the right by i elements

Matrix Product Encoding

- The simpler way to encode a codeword is by multiplying the source word, of length k, by the generator matrix (GM) of the code, obtaining a codeword of length n
- Since the LDPC codes are defined by the parity check matrix (PCM), a preprocessing to compute the generator matrix is required.
    - The computational cost of this operation is $O(n^3)$
    - The computed matrix can then be stored in the local memory so that this operation need to be done only once
- Unfortunately the resulting generator matrix is not sparse any more
    - Because of this the matrix product computational cost is $O(n^2)$ for each codeword
    - For long codewords the encoding time can be a bottleneck

Back-Substitution Encoding I

- Exploiting the particular structure of the parity-check matrix, it is possible to encode a word without computing the generator matrix, by using a back-substitution procedure that require only the parity-check matrix
- The computational complexity of this encoding strategy is almost $O(n)$

Back-Substitution Encoding II

$$H = [H_1; H_2^{'}; H_2^{''}]$$

$$[H_1; H_2^{'}; H_2^{'}][u; p'; p''] = 0$$

$$H_1 = \begin{bmatrix} H_1^1 \\ H_1^2 \\ \cdots \\ H_1^{nmr} \end{bmatrix}$$

nmr is the number of macro rows in the parity matrix

Encoding process

1. Parity of systematic part

$$s = H_1 u \quad s_j = H_1^j u$$

2. First W parity bit

(exploit weight three column)

$$p' = \sum_{j=1}^{nmr} s_j$$

3. Update *s*

$$\tilde{s}_j = s_j + H_2^{'j} p'$$

4. Remaining parity bit

(back-substitution)

$$p''^{(j)} = \tilde{s}_j \quad for \quad j = 1$$

$$p''^{(j)} = \tilde{s}_j + p''^{(j-1)} \quad for \quad j > 1$$

MAP Symbol Detection

- The decoding of the LDPC codes is performed using the MAP symbol detector, whose target is the a posteriori probability (APP)

$$\hat{u}_l = \arg\max_{u_l} p(u_l|\mathbf{r})$$

that can be rewritten as

$$\hat{u}_l = \arg\max_{u_l} \sum_{\mathbf{u}\sim u_l} \sum_{\mathbf{s}} \sum_{\mathbf{c}} M(\mathbf{u},\mathbf{c}) \prod_i p(r_i|c_i) \prod_j p(u_j)$$

where $M(\mathbf{u},\mathbf{c})$ is the code membership function, a function that takes the value 1 when $\mathbf{c} = \mathrm{E}(\mathbf{u})$ and 0 otherwise

Graphical representation of the code

- When the function to be marginalized is factorizable it is possible to represent it in a graphical way using the *Forney Style Factor Graph* and marginalize it exploiting the sum-product algorithm on the graph

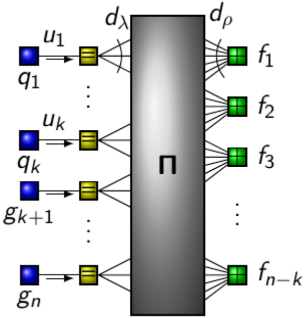

Figure : Regular LDPC FFG

Message Passing for LDPC Codes I

- Leaf Messages:

$$q_l(u_l) = \frac{1}{\sqrt{2\pi\sigma_\omega^2}} e^{-\frac{1}{2\pi\sigma_\omega^2}(r_l - M(c_l))^2} p(u_l)$$

$$g_l(c_l) = \frac{1}{\sqrt{2\pi\sigma_\omega^2}} e^{-\frac{1}{2\pi\sigma_\omega^2}(r_l - M(c_l))^2}$$

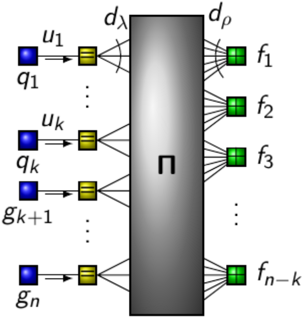

Figure : Regular LDPC FFG

Message Passing for LDPC Codes II

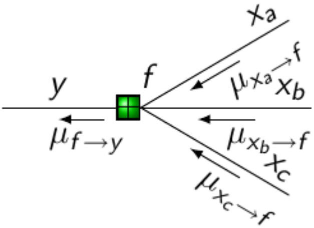

Figure : Check node

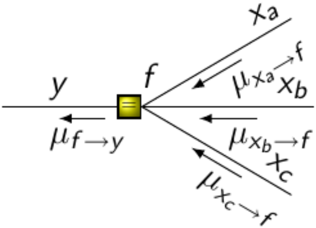

Figure : Variable node

Message Passing for LDPC Codes III

- Variable nodes message

$$\mu_{f\to y}(y) = \sum_{x_a,x_b,x_c} \delta_{x_a,y}\delta_{x_b,y}\delta_{x_c,y}\mu_{x_a\to f}(x_a)\mu_{x_b\to f}(x_b)\mu_{x_c\to f}(x_c)$$
$$= \mu_{x_a\to f}(y)\mu_{x_b\to f}(y)\mu_{x_c\to f}(y)$$

- Check nodes message

$$\mu_{f\to y}(y) = \sum_{x_a,x_b,x_c} \delta_{x_a,y+x_b+x_c}\mu_{x_a\to f}(x_a)\mu_{x_b\to f}(x_b)\mu_{x_c\to f}(x_c)$$
$$= \sum_{x_b,x_c} \mu_{x_a\to f}(y+x_b+x_c)\mu_{x_b\to f}(x_b)\mu_{x_c\to f}(x_c)$$

Message Passing for LDPC Codes IV

- Initialization

$$\mu_{c_i \to \pi} = 1$$

- Scheduling
  - Flooding: Update all the factor nodes at the same time
- Stop conditions
  - Stop when all the check nodes give 1 as output
  - Stop after N iterations
- Output

$$\hat{u}_l = \arg\max_{u_l} q_l(u_l)\mu_{f \to u_l}(u_l)$$

Message Passing for LDPC Codes - Logarithmic version I

- It is possible to implement the message passing decoder in a much more efficient way exploiting the concept of Log Likelyhood Ratio (LLR) and using some particular functions. Infact as we'll see some equation will take a simpler form do to the fact that the logarithm let the multiplication become a sum
- Log Likelyhood Ratio (LLR)

$$LLR_i = ln\frac{\mu_y(0)}{\mu_y(1)}$$

Message Passing for LDPC Codes - Logarithmic version II

- Variable nodes message

$$LLR_y = \sum_i LLR_i$$

- Check nodes message

$$LLR_y = \prod_i sign(LLR_i)\tilde{\phi}\left(\sum_i \tilde{\phi}(|LLR_i|)\right)$$

where

$$\tilde{\phi}(x) = -ln\ \phi(x) = -ln\ tanh\left(\frac{x}{2}\right) = -ln\left(\frac{e^x - 1}{e^x + 1}\right)$$

Message Passing for LDPC Codes - Logarithmic version III

- Leaf messages

$$LLRg_l \rightarrow u_l = -\frac{2r_l}{\sigma_\omega^2}$$

- Output

$$u_l = \begin{cases} 0 & \text{if } LLR_{g_l \rightarrow u_l} + \sum_{i \in A_l} LLR_i > 0 \\ 1 & \text{otherwise} \end{cases}$$

where

$A_l =$ Set of check nodes in which the variable node l is involved

Message Passing for LDPC Codes - Logarithmic version IV

Algorithm 4 Sum-Product Decoding

procedure DECODE($\mathbf{r}$)

  $I = 0$ ▷ Initialization

  for $i = 1 : n$ **do**

    for $j = 1 : m$ **do**

      $M_{j,i} = r_i$

    end for

  end for

  repeat

    for $j = 1 : m$ **do** ▷ Step 1: Check messages

      for $i \in B_j$ **do**

        $E_{j,i} = \log\left(\frac{1+\prod_{i' \in B_j, i' \neq i} \tanh(M_{j,i'}/2)}{1-\prod_{i' \in B_j,\, i' \neq i} \tanh(M_{j,i'}/2)}\right)$

      end for

    end for

Figure : BER and FER vs Eb/N0, Rate 1/2

Message Passing for LDPC Codes - Logarithmic version V

for $i = 1 : n$ **do** ▷ Test
$L_i = \sum_{j \in A_i} E_{j,i} + r_i$
$z_i = \begin{cases} 1, & L_i \leq 0 \\ 0, & L_i > 0. \end{cases}$
end for
if $I = I_{\max}$ or $H\mathbf{z}^T = 0$ **then**
Finished
else
for $i = 1 : n$ **do** ▷ Step 2: Bit messages
for $j \in A_i$ **do**
$M_{j,i} = \sum_{j' \in A_i,\, j' \neq j} E_{j',i} + r_i$
end for
end for
$I = I + 1$
end if
until Finished
end procedure

Figure : BER and FER vs Eb/N0, Rate 1/2

Simulations

- To verify the correctness of the MATLAB implementation of the 802.11n LDPC encoding system, some simulations have been carried out. The parameters used in the simulations are the following:

Simulation parameters I

- Simulation parameters common to each simulations
  - BPAM modulation
  - AWGN channel model
  - Packet length n=1944 bits (maximum available length) with subblock dimension of W=81 bits
  - 40 message passing iterations
- Simulation 1 - BER and FER vs $\frac{E_b}{N_0}$ for rate $\frac{1}{2}$ code
  - $\frac{E_b}{N_0} \in [0.75, 2.5]$
  - Input length 10^7
  - Results averaged over 10 simulations

Simulation parameters II

- Simulation 2 - BER and FER vs number of message passing iterations for rate $\frac{1}{2}$ code
  - $\frac{E_b}{N_0} = 1.5$
  - Input length 10^5
  - Results averaged over 1000 simulations
- Simulation 3 - BER and FER vs $\frac{E_b}{N_0}$ comparison for all the different code rates available
  - $\frac{E_b}{N_0} \in [0.75, 4.5]$
  - Input length 10^7
  - Results averaged over 10 simulations

BER and FER vs Eb/N0, Rate 1/2

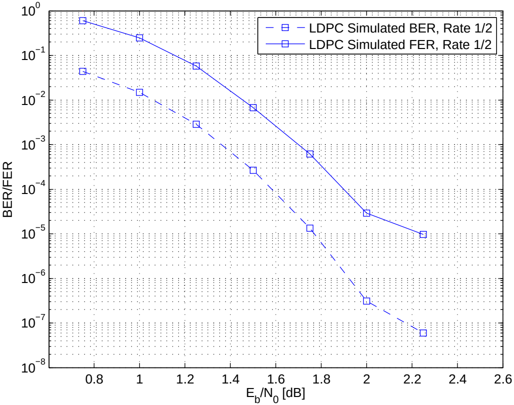

Figure : BER and FER vs Eb/N0, Rate 1/2

BER and FER vs Iterations, Rate 1/2

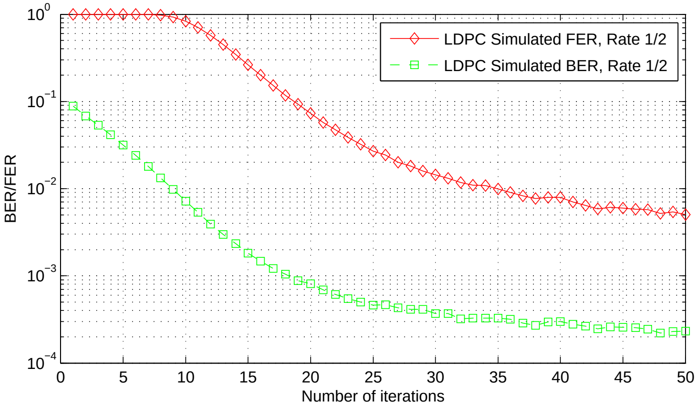

Figure : BER and FER vs Iterations, Rate 1/2

BER vs Eb/N0, Code comparison

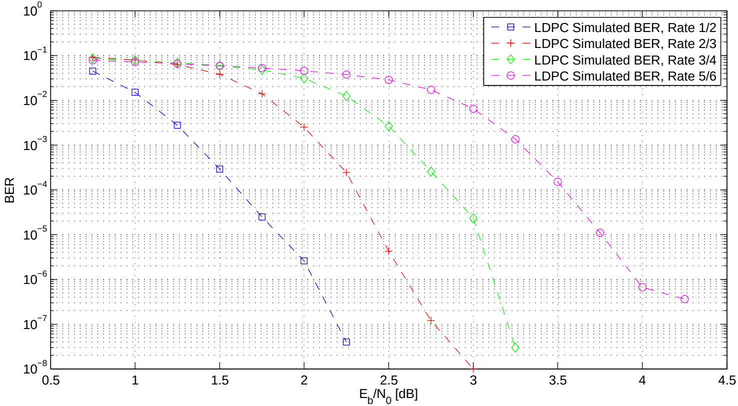

Figure : BER vs Eb/N0, Code comparison

FER vs Eb/N0, Code comparison

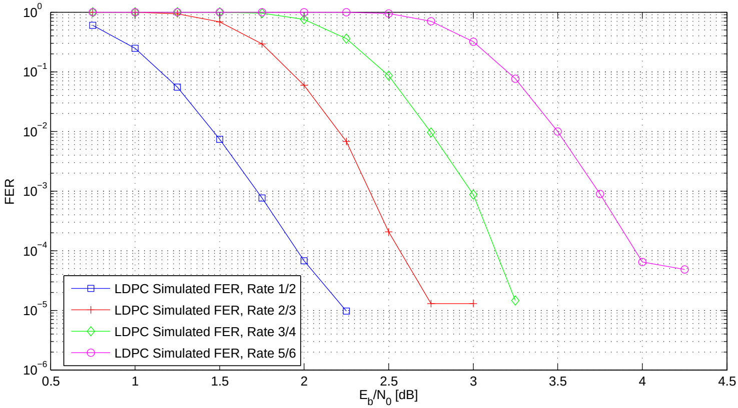

Figure : FER vs Eb/N0, Code comparison

Conclusions I

- As we have seen from the simulations the LDPC codes can lead to a great improvement in the performance in term of BER compared to other classical coding system as convolutional codes
- In the second simulation we see that the performance increase with the number of iterations, but we also see that after a certain number of iterations, about 25, the improvement is not very significant while the decoding time increase linearly with the number of iterations. This means that the maximum number of iterations must be carefully chosen to find the best tradeoff between BER performance and decoding time

Conclusions II

- Another important aspect to be taken into account is the choice of the parity check matrix. A good designed parity check matrix can give better performance in terms of BER and minimize the complexity of the encoding process, and finally minimize the encoding time

Bibliography

- 802.11-2012 – IEEE Standard for Information technology - Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications
- *Raffaele Riva* – LDPC codes for W-LAN IEEE 802.11n: An overview – LDPC at work, 2006
- *Pascal Urard, Stefano Valle* – Digital implementations of Low-Density Parity-Check Codecs - 5th Analog Decoding Workshop, Turin, Italy, June 5-6, 2006
- *Andres I. Vila Casado, Miguel Griot, and Richard D. Wesel, Senior Member, IEEE* – LDPC Decoders with Informed Dynamic Scheduling