Random Construction을 이용한 LDPC 부호의 성능 분석

2013 한국 통신학회 동계학술대회

2013/01/31

전석민, **김인선**, 박진수, 박기현, 송홍엽

Contents

Introduction

- LDPC codes are represented by check and variable node connection

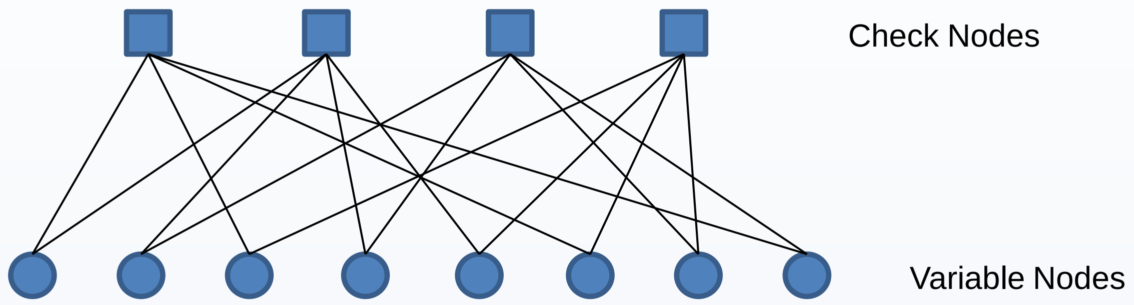


Fig.1 Tanner graph of LDPC codes

- Degree Distribution determines the number of edges of Tanner graph
  - Choosing good degree distribution is important due to good performance

    $\lambda(x)$: *check node degree distribution* $\rho(x)$: *variable node distribution*

- There are two kinds degree distribution of LDPC code
  - Regular LDPC code, Irregular LPDC code

Introduction

- LDPC codes construction
  - PEG, ACE, Random …

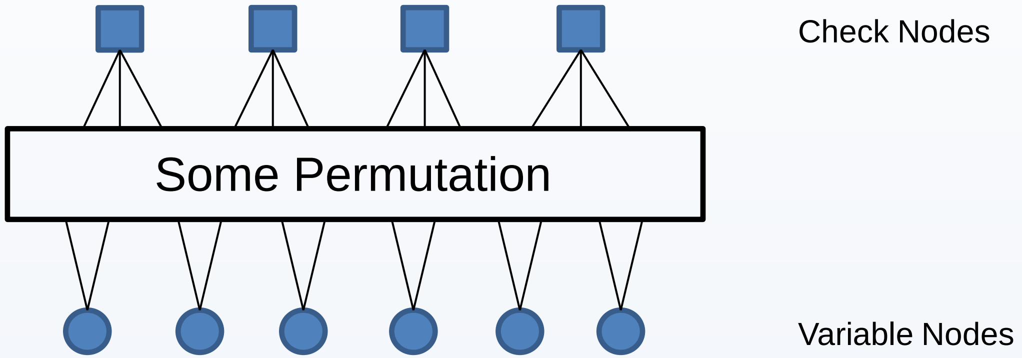


Fig.2 Tanner graph representation of parity-check matrix

- Good BER performance can be obtained from some random parity check matrix with sufficient length (Richardson 2001)
- Suggest that simple permutation method & parity check matrix construction

Introduction

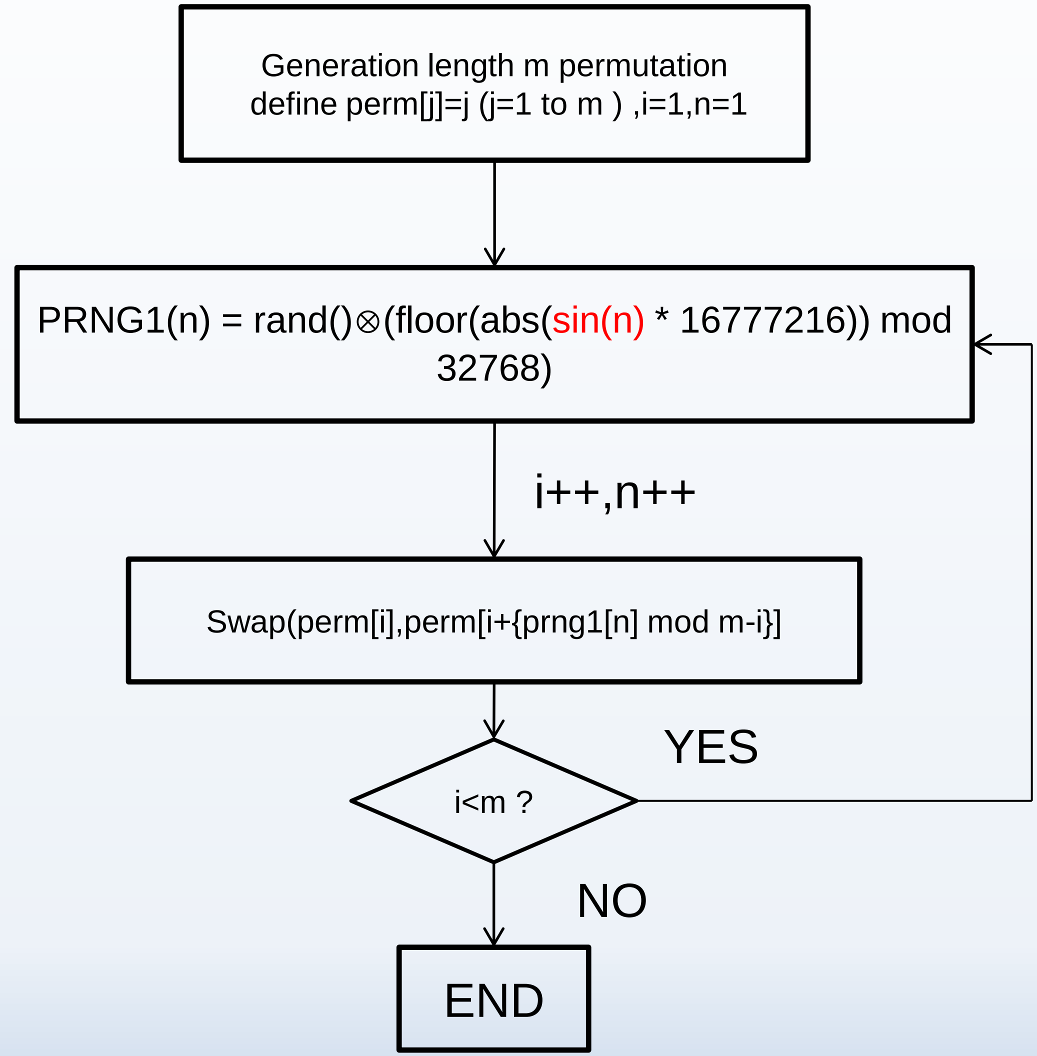
- Research objectives
  - **Find sufficient codeword length** which can achieve good BER performance
  - **Verify the performance** of randomly constructed regular and irregular codes of lengths around 100, 500, 1000, 2000, and 4000.
  - BER performance of the codes is compared with that of some PEG-optimized codes in 802.11n standard and 802.16 standard.
  - **Result Analysis** why they are a bit worse.

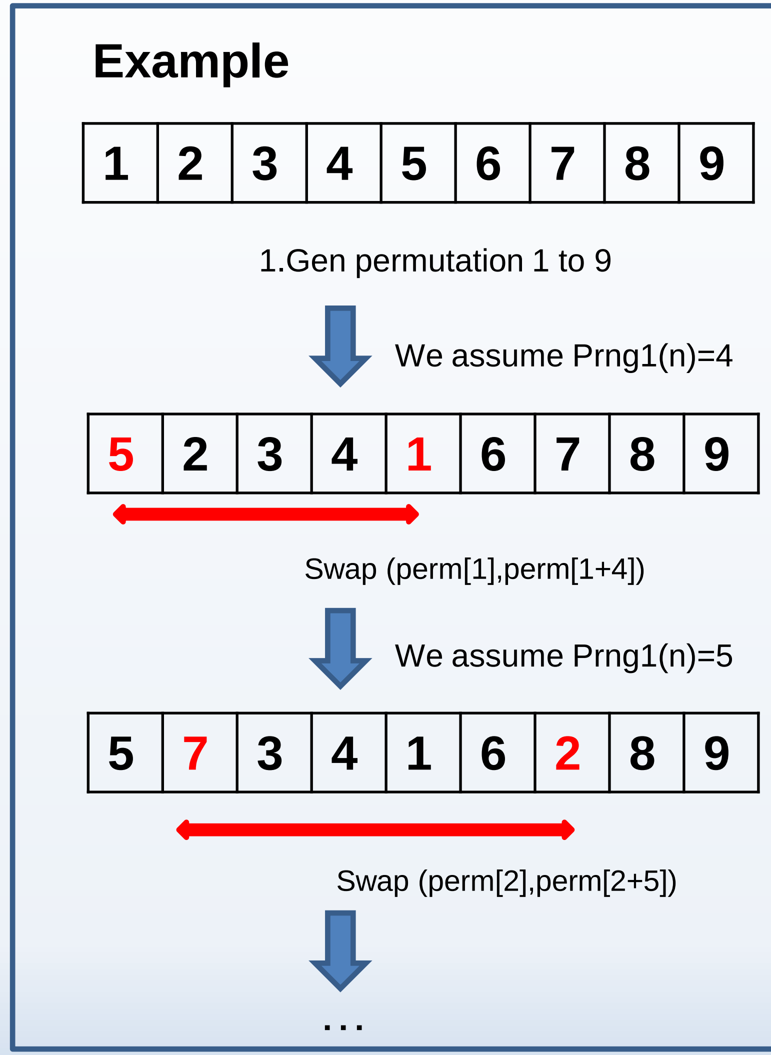
Random Permutation

PRNG1	$PRNG(n) = rand()\^{}(floor(abs(\sin(n) * 16777216))mod\ 32768)$ ^:Bit-wise XOR
PRNG2	$prem[i] = floor(M \times \sin[n \times x])$
Quadratic hash	$x(i+1) = [x(i) + 2 * i + 1]\ (mod\ m)$ $x(i+1) = [x(i) + t](mod\ m)$
Primitive root	The primitive root is defined by p_1 $p_1{}^i = n(mod\ m)$

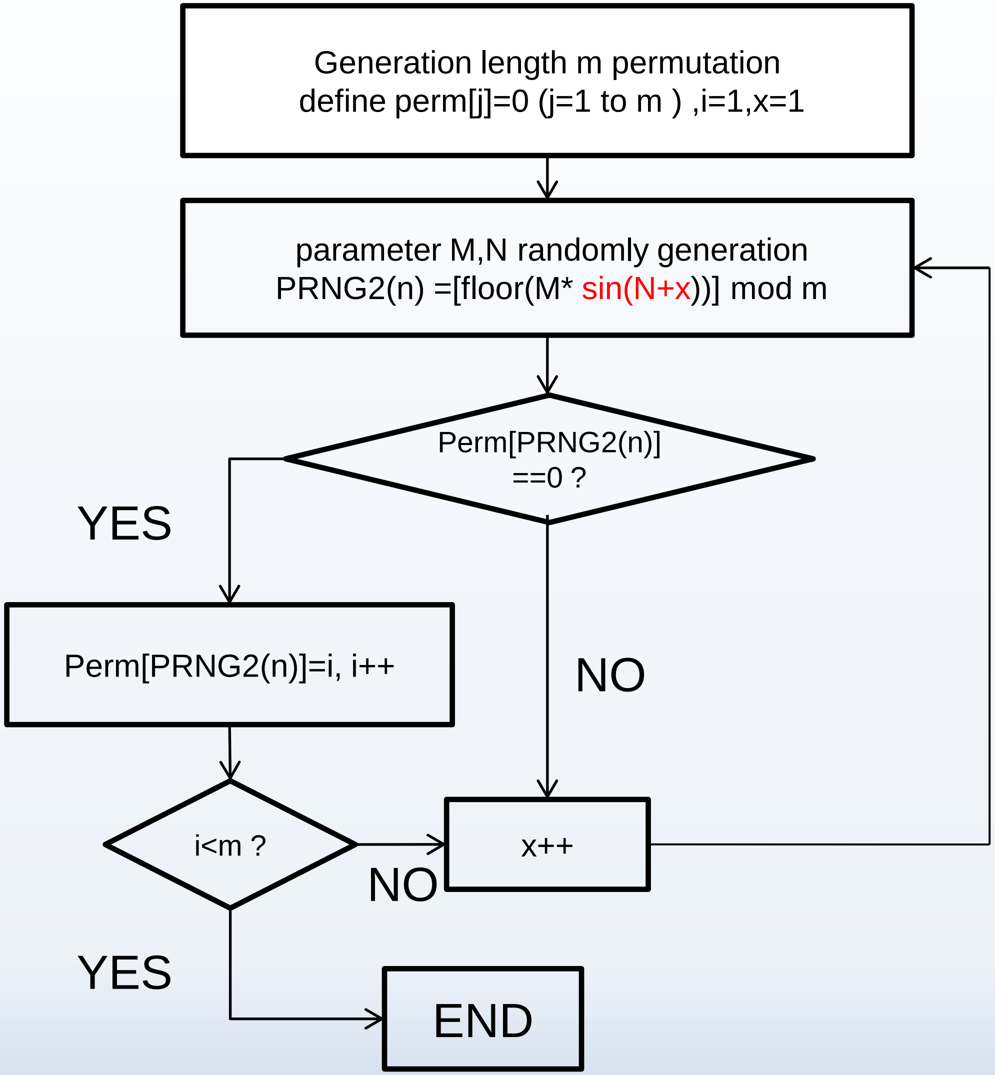
Generation of Random Permutation

1. Permutation generated using the non-periodicity of the sine function(PRNG1)





2. Permutation generated using the non-periodicity of the sin function(PRNG2)



Example

0	0	0	0	0	0	0	0	0

Generate perm[j]=0 (j=1 ~m)

↓ perm[1]=5

0	0	0	0	1	0	0	0	0

↓ perm[2]=3

0	0	2	0	1	0	0	0	0

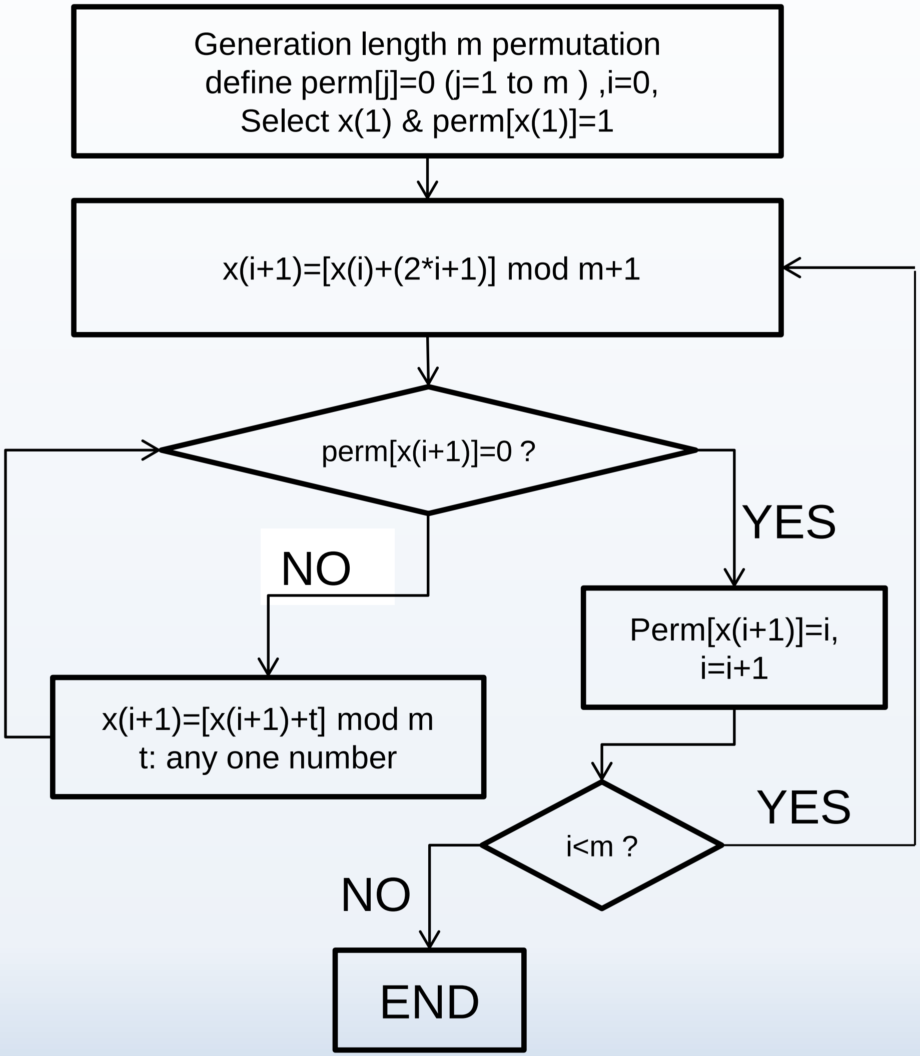
↓ perm[3]=5

Regeneration PRNG2(n)

…

Generation of Random Permutation

- 3. Quadratic hash permutation



Example

X(1)=3

0	0	0	1	0	0	0	0	0

X(2)=(3+2*1+1) mod 9

0	0	0	1	0	0	2	0	0

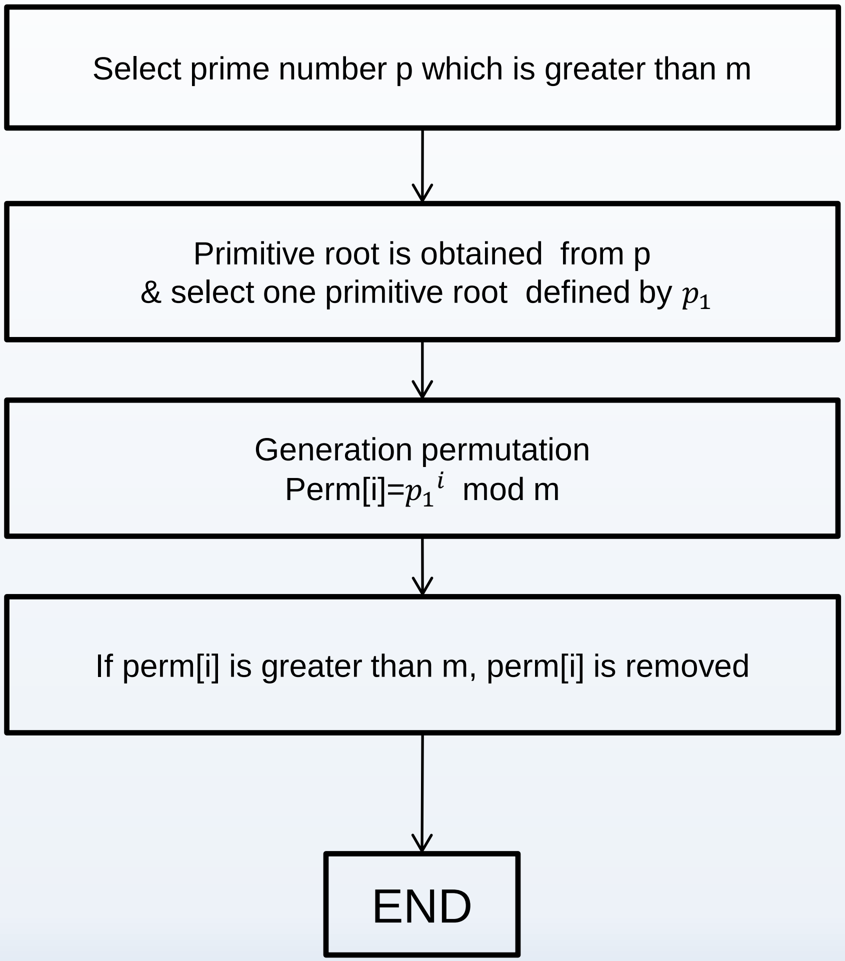
X(3)=(6+2*2+1) mod 9

0	0	3	1	0	0	2	0	0

…

Generation of Random Permutation

4. Primitive root permutation



Example

Select number 3 which is primitive of 5

3 → 3
4 → 4=(3*3) mod 5
2 → 2=(4*3) mod 5
1 → 1=(2*3) mod 5

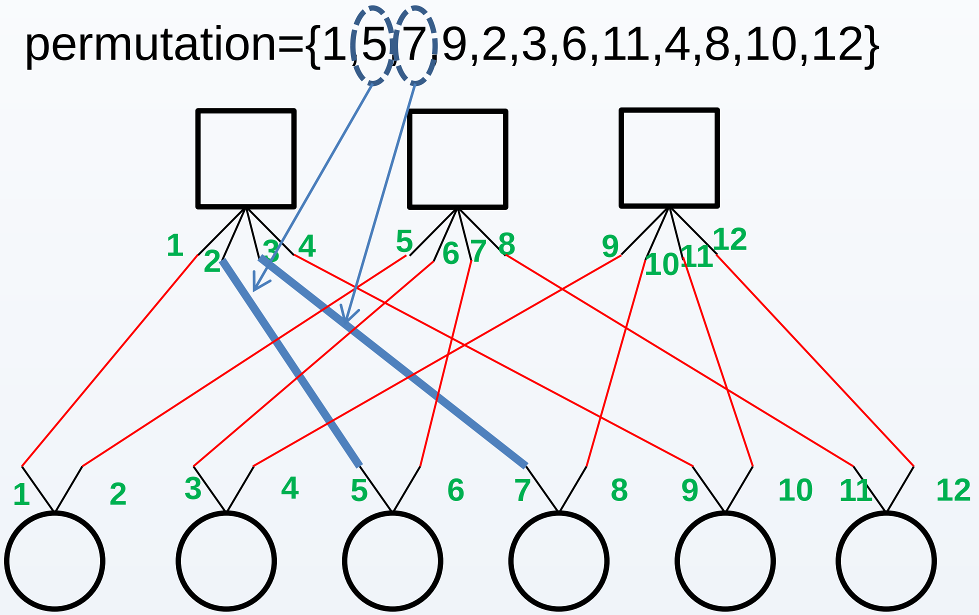
2 → 2
4 → 4=(2*2) mod 5
3 → 3=(4*2) mod 5
1 → 1=(3*2) mod 5

Generation of Parity-Check Matrix

- Method of generation of parity check matrix by some permutation
  - Example

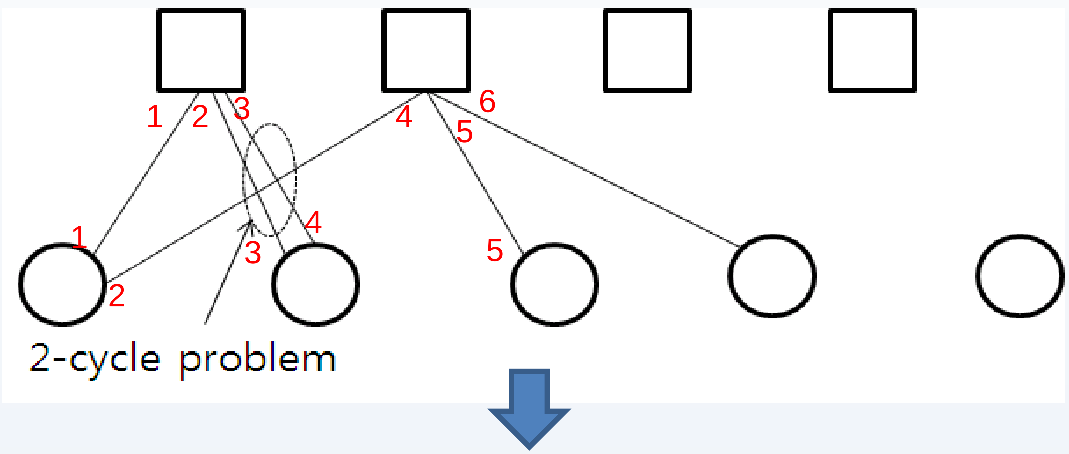
    (2,4) regular LDPC code

    permutation={1,5,7,9,2,3,6,11,4,8,10,12}



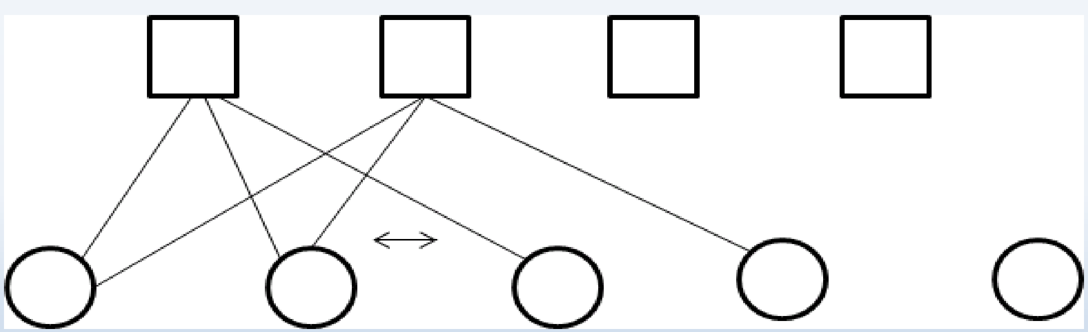
Generation of Parity-Check Matrix

- 2-Cycle problem
  - Two lines from one check node can connect to one variable node
  - If the problem appears, we regenerated the permutation or exchanged the line for the line of next node



Permutation={1,3,4,2,5,7...}

2-cycle problem



Permutation={1,3,5,2,4,7...}

Simulation Settings

- Regular LDPC codes of the proposed method are compared with codeword length100,500,1000,2000 (3,6) ,(4,8)regular PEG-optimized LDPC code.
- Irregular LDPC codes of the proposed method are compared with PEG-optimized and 802.11n standard LDPC code which length is 648, 1296, 2284, 4568 in several codes on the standard.(Rate:0.5)
- Degree distribution of irregular LDPC code is

$$\lambda(x) = 0.2558x + 0.3140x^2 + 0.0465x^3 + 0.3837x^{10}$$
$$\rho(x) = 0.8140x^6 + 0.1860x^7$$

variable node				check node	
# degree 2	# degree 3	# degree 4	# degree 11	# degree 7	# degree 8
594	486	54	162	540	108

Length 1296 802.16n LDPC standard code degree

Simulation Settings

- Irregular LDPC codes of the proposed method are compared with PEG-optimized and 802.16 standard LDPC code which length is 576, 1056, 2016 in several codes on the standard.(Rate:0.5)
- Degree distribution of irregular LDPC code is

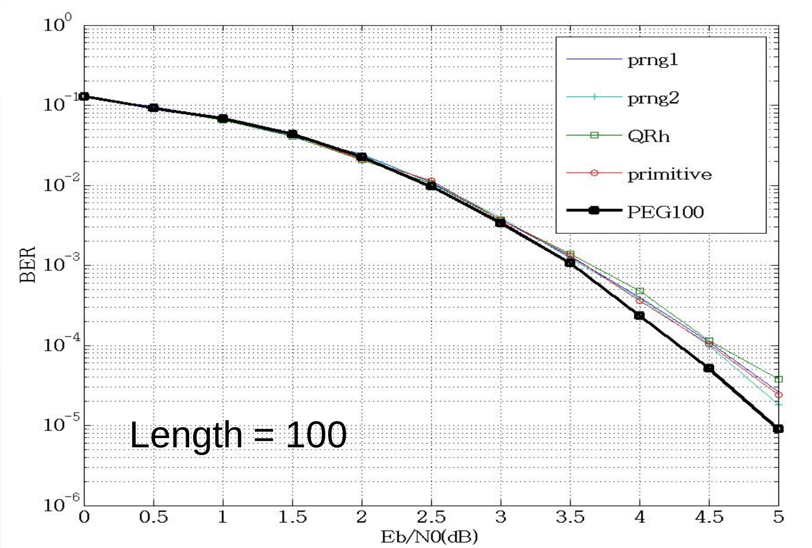
$$\lambda(x) = 0.4583x + 0.3333x^2 + 0.2083x^3$$
$$\rho(x) = 0.6667x^6 + 0.3333x^7$$

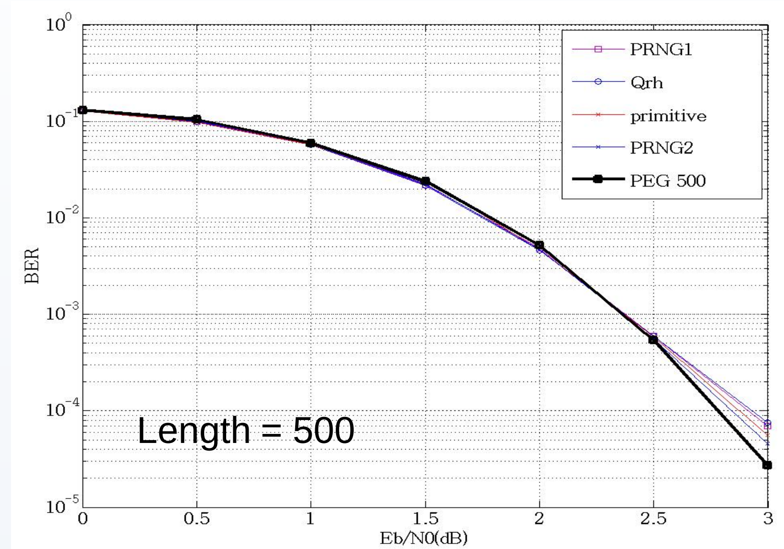
variable node			check node	
# degree 2	# degree 3	# degree 6	# degree 7	# degree 8
484	352	220	352	176

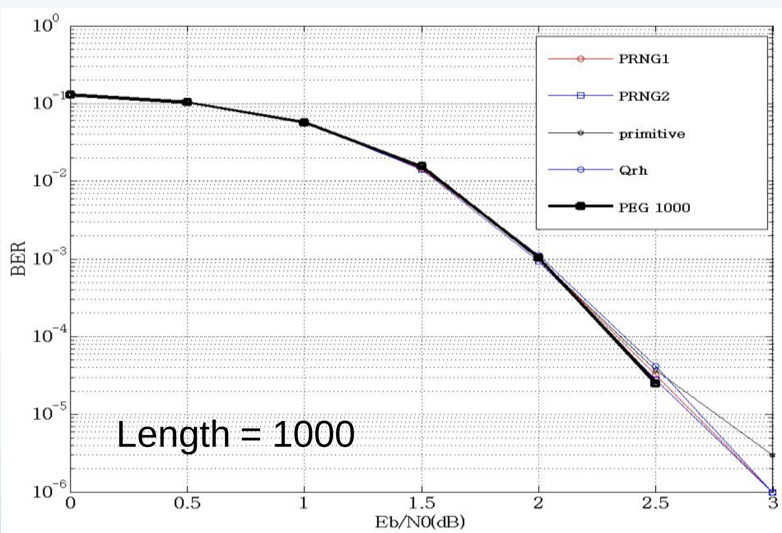
Length 1056, 802.16n LDPC standard code degree

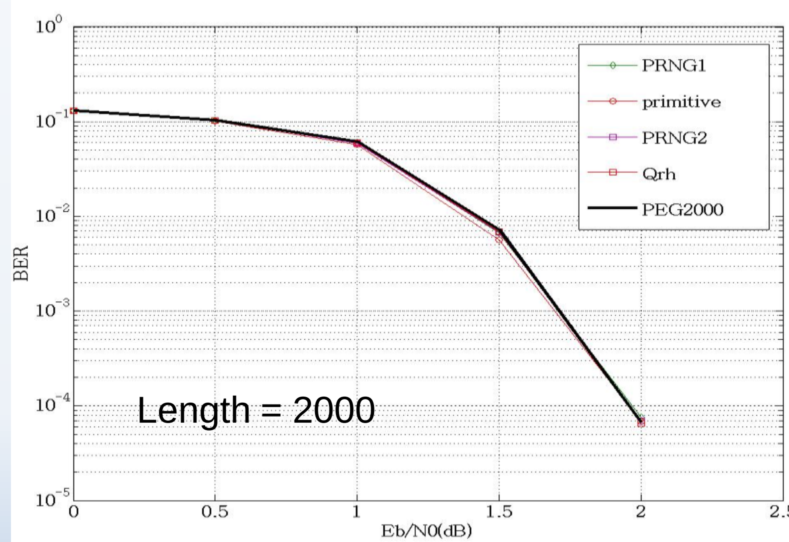
Simulation Result

- (3,6) Regular LDPC Codes – Length 100,500,1000,2000



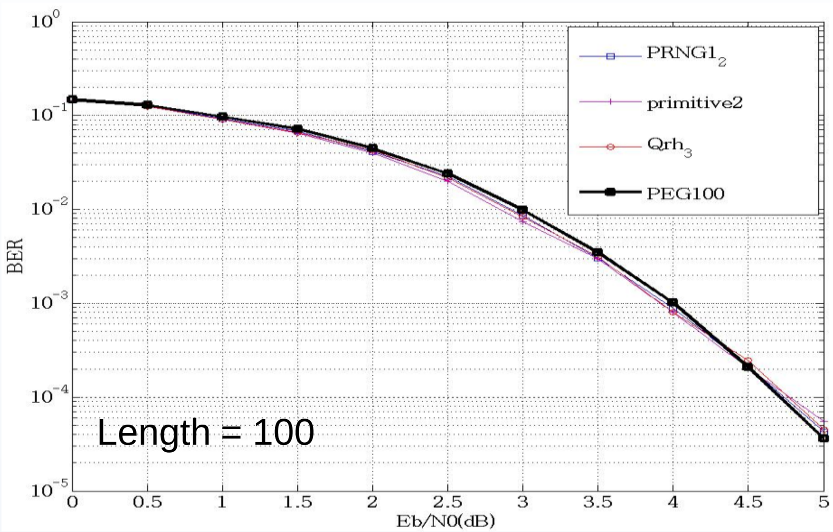


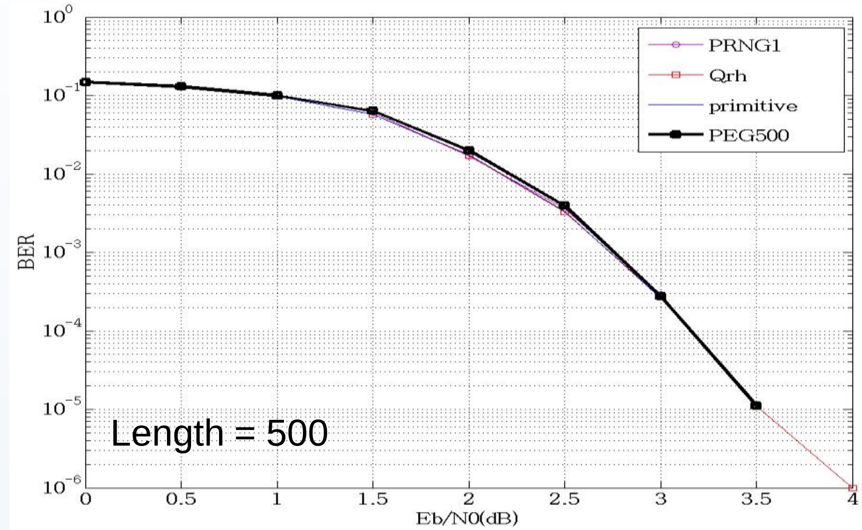


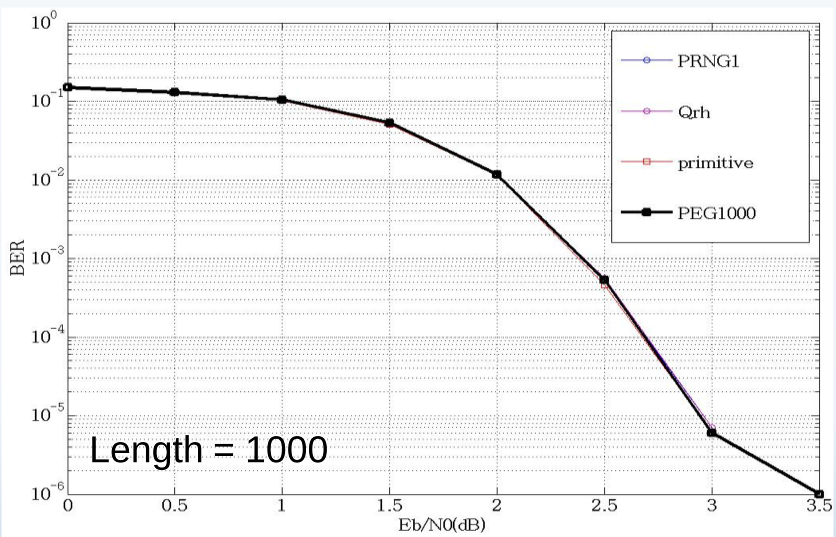


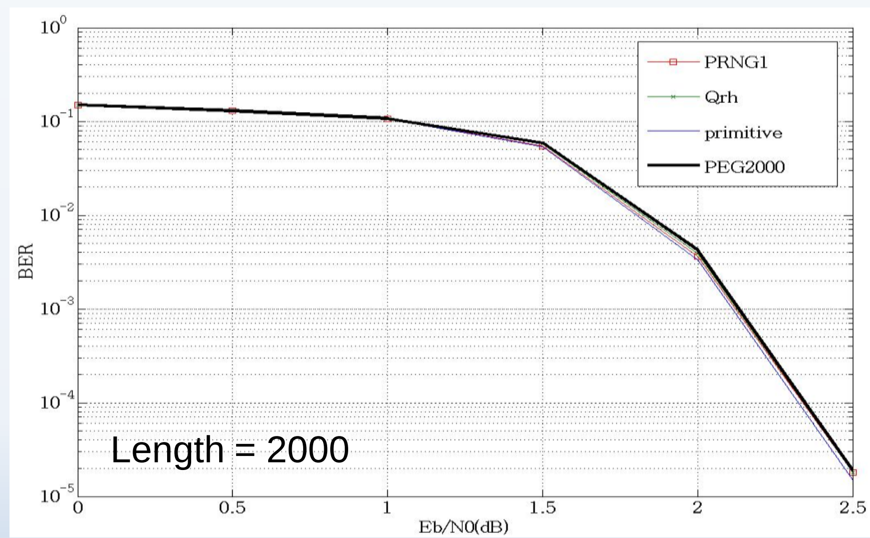
Simulation Result

- (4,8) Regular LDPC Codes – Length 100,500,1000,2000



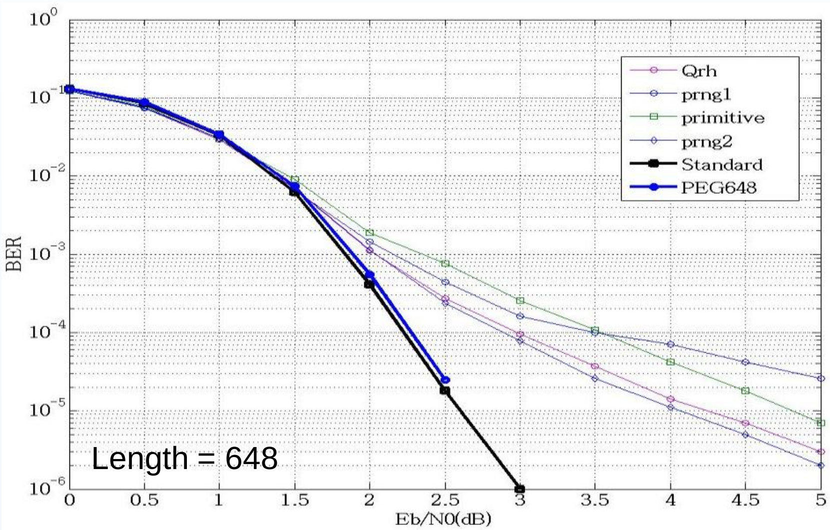


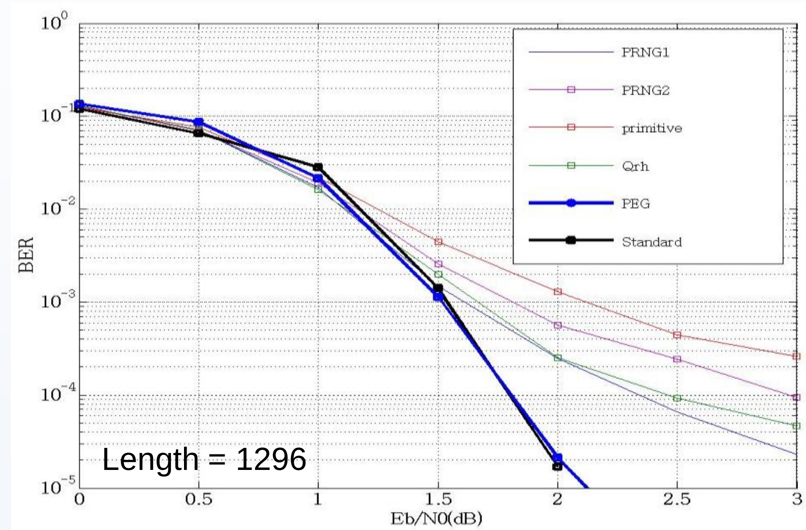


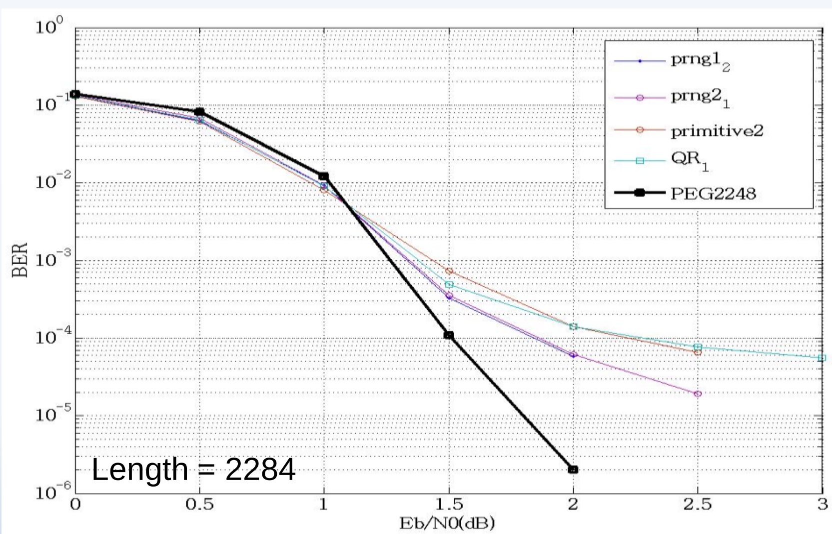


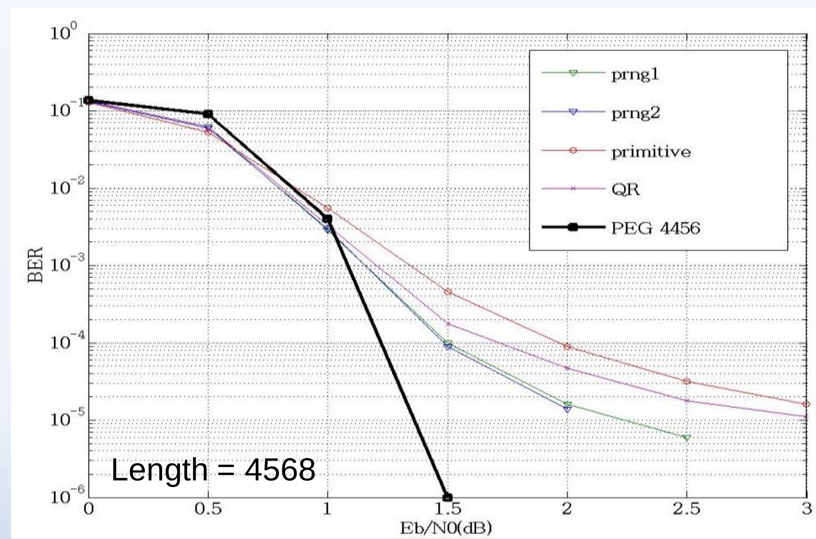
Simulation Result

- 802.11n Standard LDPC Codes(Irregular) – Length 648, 1296, 2284, 4568



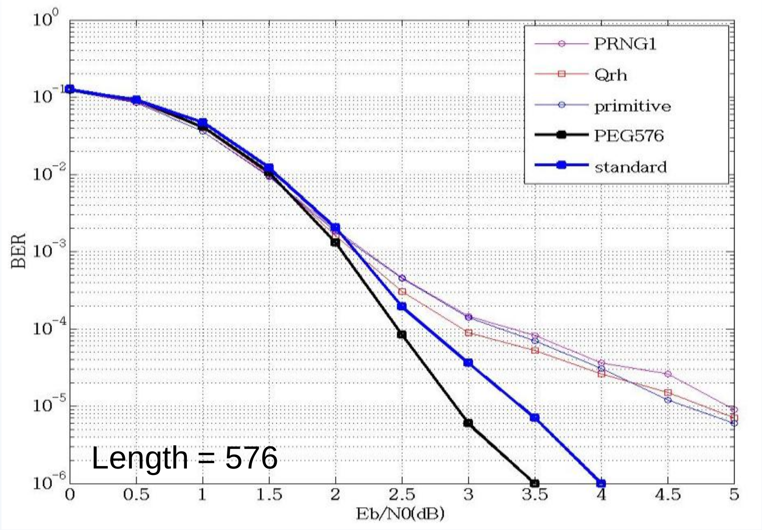


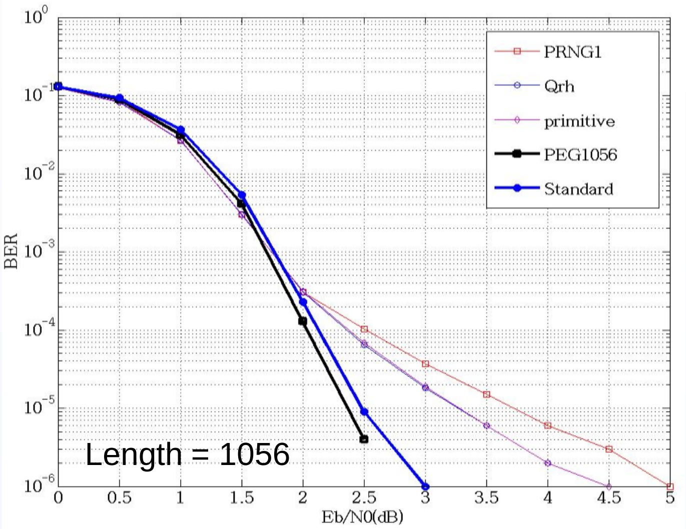


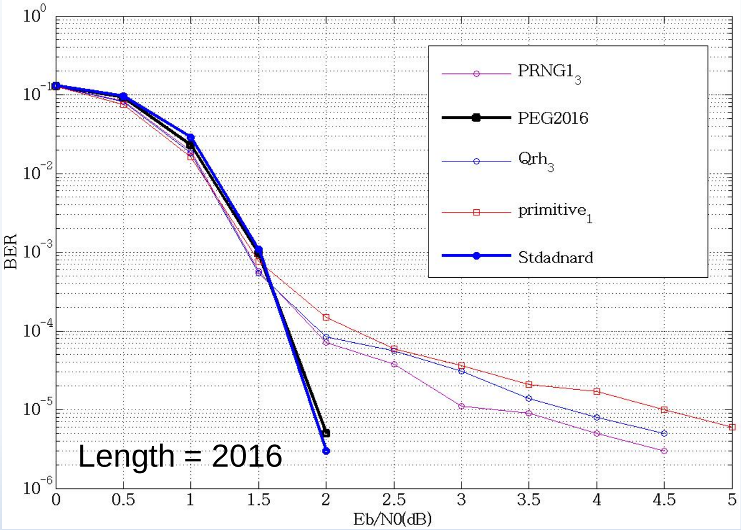


Simulation Result

- 802.16 Standard LDPC Codes – Length 576, 1056, 2016

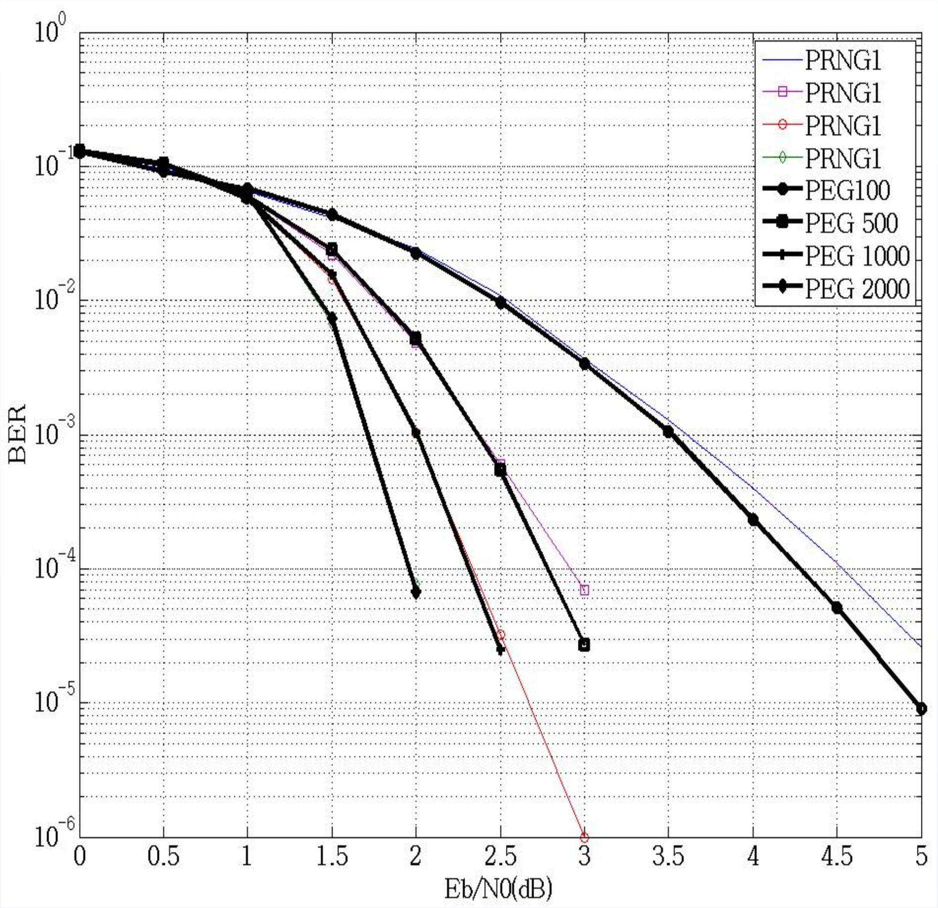




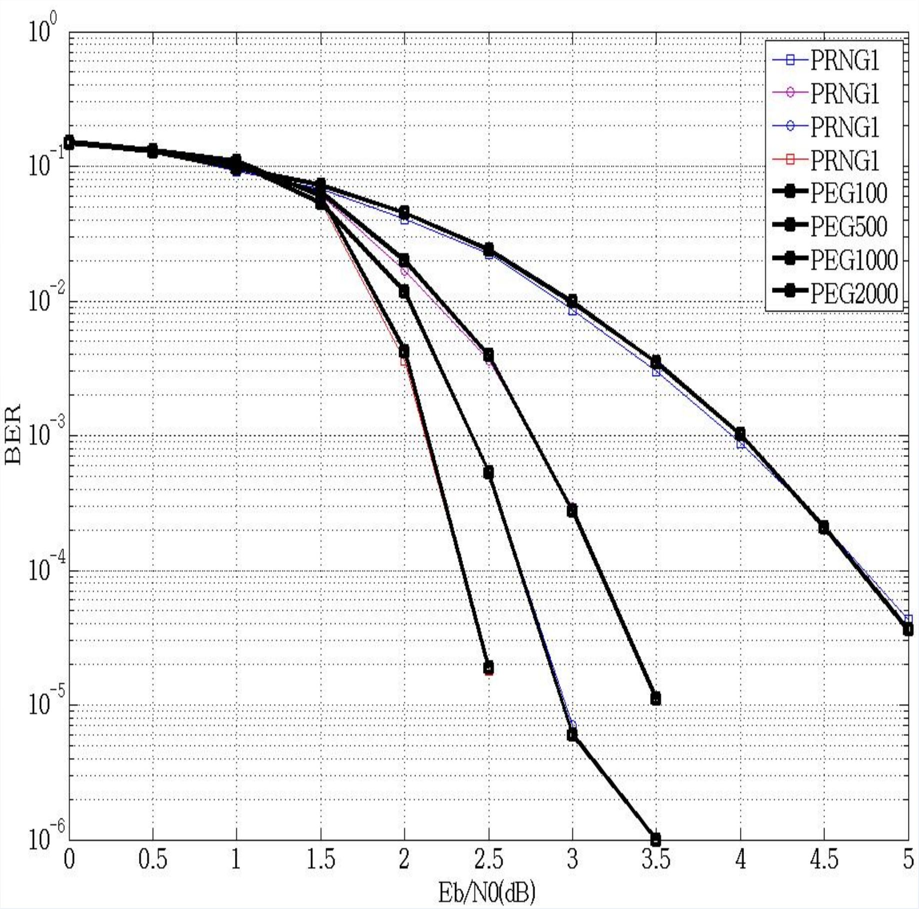


Simulation Result

- (3,6) & (4,8) Regular LDPC Codes Simulation Result



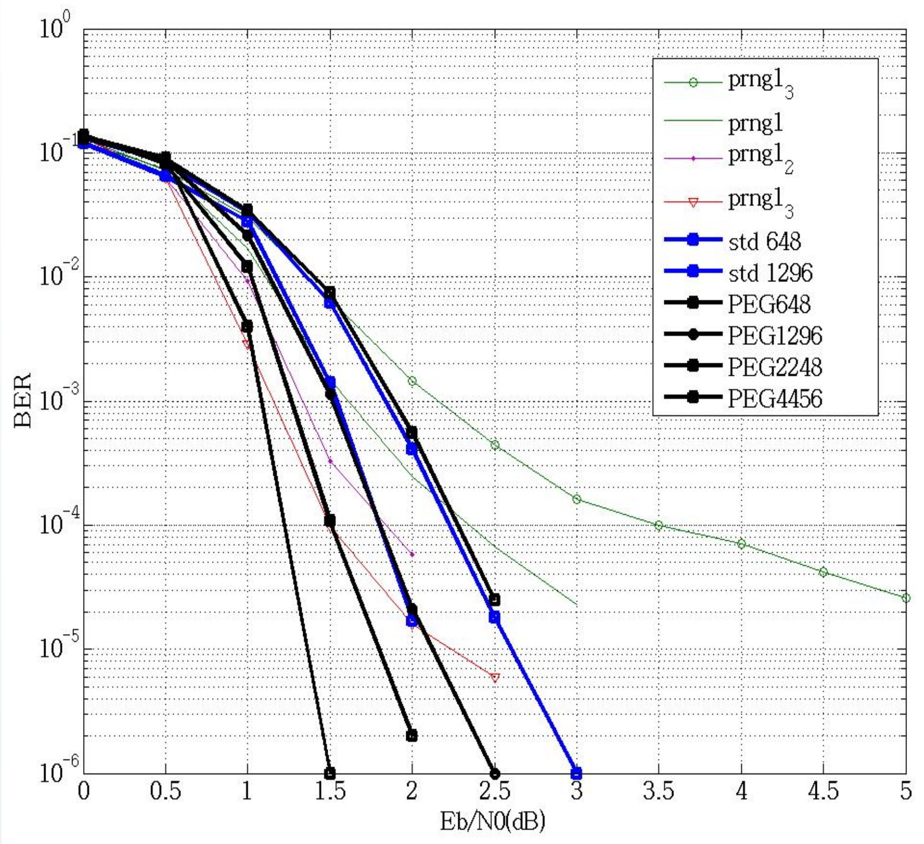
BER performance of (3,6) Regular LDPC Codes



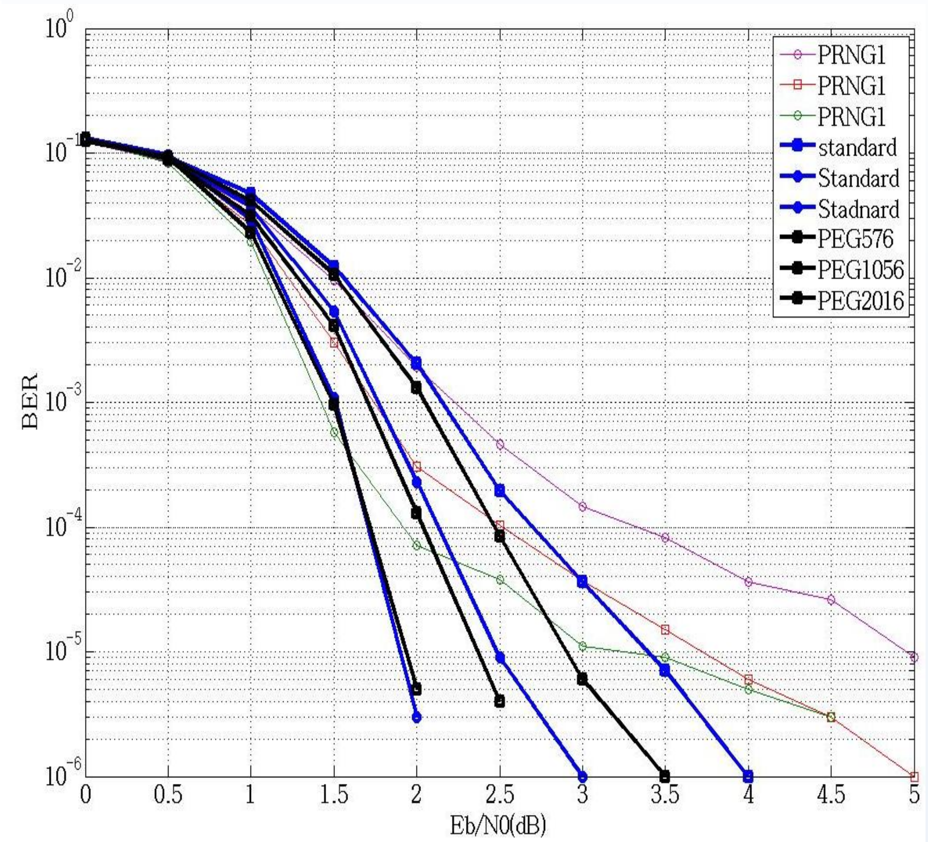
BER performance of (4,8) Regular LDPC Codes

Simulation Result

- 802.11n & 802.16 Irregular LDPC Codes Simulation Result



BER performance of 802.11n Irregular LDPC Codes



BER performance of 802.16 Irregular LDPC Codes

Conclusions

- We can achieve good BER performance by using random permutation in waterfall region without some optimization process
- In (3,6) regular code, the proposed codes has same BER performance as PEG optimized code at codeword length 2000.
- In (4,8) regular code, the proposed codes has better performance than PEG optimized code at codeword length 100.
- The BER simulation results show that the SNR loss between proposed method and standard LDPC code in irregular LDPC code