

802.11n 규격에서의 저복잡도 LDPC 복호 알고리즘

준회원 김민혁*, 박태두*, 정회원 정지원*, 이성로**, 정민아**

Low Computational Complexity LDPC Decoding Algorithms for 802.11n Standard

Min-hyuk Kim*, Tae-doo Park* *Associate Members*,
Ji-won Jung*, Seong-ro Lee**, Min-a Jung *Regular Members*

요약

본 연구에서는 무선 랜 표준안인 802.11n에서 채널 부호화 알고리즘으로 채택된 LDPC부호의 복호 알고리즘의 저복잡도에 대해 연구를 하였다. 사본의 한계에 근접하기 위해서는 큰 블록 사이즈의 LDPC 부호어 길이와 많은 반복횟수를 요구한다. 이는 많은 계산량을 요구하며, 그리고 이에 따른 전력 소비량(power consumption)을 야기 시키므로 본 논문에서는 세 가지 형태의 저복잡도 LDPC 복호 알고리즘을 제시한다. 첫째로 큰 블록 사이즈와 많은 반복 횟수는 많은 계산량과 전력 소모량을 요구하므로 성능 손실 없이 반복횟수를 줄일 수 있는 부분 병렬 방법을 이용한 복호 알고리즘, 둘째로 early stop 알고리즘에 대해 연구 하였고, 셋째로 비트 노드 계산과 체크 노드 계산 시 일정한 신뢰도 값보다 크면 다음 반복 시 계산을 하지 않는 early detection 알고리즘에 대해 연구 하였다.

Key Words : Low Density Parity Check Code, Early stop algorithm, Sequential decoding, Early detection algorithm

ABSTRACT

In this paper, we first review LDPC codes in general and a belief propagation algorithm that works in logarithm domain. LDPC codes, which is chosen 802.11n for wireless local access network(WLAN) standard, are required a large number of computation due to large size of coded block and iteration. Therefore, we presented three kinds of low computational algorithm for LDPC codes. First, sequential decoding with partial group is proposed. It has same H/W complexity, and fewer number of iteration's are required at same performance in comparison with conventional decoder algorithm. Secondly, we have apply early stop algorithm. This method is reduced number of unnecessary iteration. Third, early detection method for reducing the computational complexity is proposed. Using a confidence criterion, some bit nodes and check node edges are detected early on during decoding. Through the simulation, we knew that the iteration number are reduced by half using subset algorithm and early stop algorithm is reduced more than one iteration and computational complexity of early detected method is about 30% offs in case of check node update, 94% offs in case of check node update compared to conventional scheme.

※ “본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음” (NIPA-2009-C1090-0902-0010), “이 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. D00280).”

* 한국해양대학교 전파공학과 위성통신 연구실(hyuk6030@hhu.ac.kr), ** 목포대학교

논문번호 : KICS2009-07-301, 접수일자 : 2009년 7월 20일, 최종논문접수일 : 2010년 1월 25일

리터 검사 행렬의 가로 길이를 n_b 라 하고 세로의 길이를 m_b 라 하면, $k_b = n_b - m_b$ 가 된다. 또한, 데이터의 집합 \mathbf{m} 이 부호화 과정을 거친 후의 데이터를 c_b 라하면 $c_b = [m_0, m_1, \dots, m_{k_b-1}, p_0, p_1, \dots, p_{m_b-1}]$, 로 표현 할 수 있다.

결과적으로 부호화가 $H \cdot c_b = 0$ 의 특징을 가지므로, 다음과 같이 표현할 수 있다.

$$\begin{bmatrix} h_{0,0} & \dots & h_{0,k_b-1} & 1 & \dots & -1 \\ h_{1,0} & \dots & h_{1,k_b-1} & -1 & \dots & \vdots \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ h_{x,0} & \dots & h_{x,k_b-1} & 0 & \dots & \vdots \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ h_{m_b-1,0} & \dots & h_{m_b-1,k_b-1} & 1 & \dots & 0 \end{bmatrix} X \begin{bmatrix} m_0 \\ \vdots \\ m_{k_b-1} \\ p_0 \\ \vdots \\ p_{m_b-1} \end{bmatrix} = 0$$

위 식을 계산하면 다음과 같다.

$$\begin{aligned} \sum_{j=0}^{k_b-1} h_{0,j} m_j + \prod_1 p_0 + p_1 &= 0 \quad (\text{0th row}) \\ \sum_j h_{i,j} m_j + p_i + p_{i+1} &= 0 \quad i \neq 0, x, m_b - 1 \\ \sum_j h_{x,j} m_j + p_0 + p_x + p_{x+1} &= 0 \quad (\text{xth row}) \\ \sum_j h_{m_b-1,j} m_j + \prod_1 p_0 + p_{m_b-1} &= 0 \quad ((m_b - 1) \text{th row}) \end{aligned} \quad (1)$$

위의 네 식을 모두 더하면,

$$p_0 = \sum_{i=0}^{m_b-1} \sum_{j=0}^{k_b-1} h_{i,j} m_j = \sum_{i=0}^{m_b-1} \lambda_i, \quad \lambda_i = \sum_{j=0}^{k_b-1} h_{i,j} m_j \quad (2)$$

을 얻을 수 있다. p_0 을 구하면, p_1 부터는 다음 식으로 구할 수 있다.

$$\begin{aligned} p_1 &= \lambda_0 + \prod_1 p_0, \quad p_{i+1} = \lambda_i + p_i (i \neq 0, x, m_b - 1), \\ p_{i+1} &= \lambda_i + p_i + p_0 (i = x), \quad p_{m_b-1} = \lambda_{m_b-1} + \prod_1 p_0 \end{aligned} \quad (3)$$

그림 2는 위 식들을 이용하여 부호화기를 만든

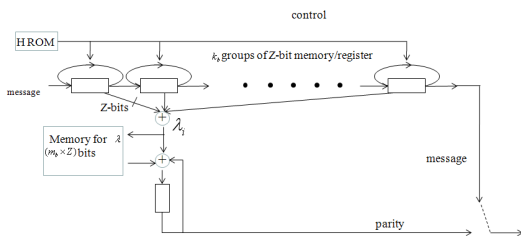


그림 2. 802.11n LDPC 부호화기의 구조

것이다. 이 구조를 이용하면 보다 쉽고 간단하게 부호화기를 설계 할 수 있다.

2.2 802.11n 시스템에서 제공하는 부호율에 대한 성능 분석

IEEE 802.11n 표준안 기반의 시스템은 4개의 서로 다른 부호화율 1/2, 2/3, 3/4 5/6을 지원하며, 각각의 부호화율에서는 세 개의 서로 다른 z 사이즈를 지원한다. 그리고 성능향상을 위해 irregular LDPC 부호를 사용한다. 802.11n에서 제공하는 각 부호화에 따른 부호화어 크기 N, 정보원의 크기 K 그리고 각 행과 열에서 ‘1’의 개수를 나타내는 row weight와 column weight의 크기는 아래 표1과 같다. 그림 3은 가우시안 잡음에서 부호화블록 사이즈 N이 1944 이고 변조 방식이 BPSK 일 때 성능 그래프이다.

표 1. 802.11n에서 제공하는 각 부호화율에 따른 파라메타

Coding rate R	Coded Word N	Information Size K	Row weight d_c	column weight d_r
1/2	648	324	8	12
	1296	648	8	11
	1944	972	8	11
2/3	648	432	11	8
	1296	864	11	8
	1944	1296	11	8
3/4	648	486	15	6
	1296	972	15	6
	1944	1458	15	6
5/6	648	540	22	4
	1296	1080	22	4
	1944	1620	20	4

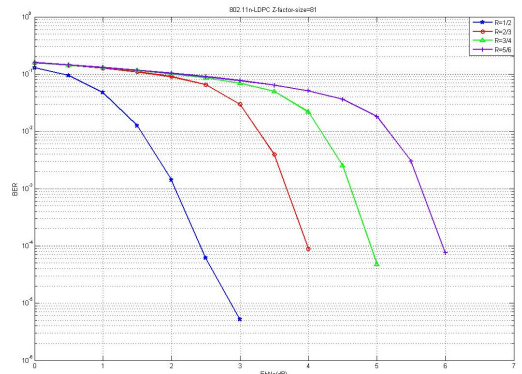


그림 3. 다양한 부호화율에서 BER 성능 곡선(반복횟수=8, N=1944, BPSK 변조방식)

III. 저복잡도 알고리즘

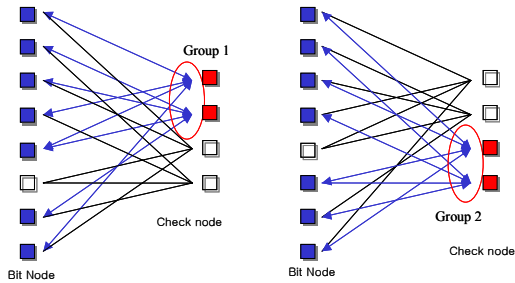
3.1 Sequential Decoding with partial Group

앞장에서 살펴보았듯이 일반적인 LDPC decoder는 bit to check block 과 check to bit block 으로 구성된다. Bit to check block 블록은 동시에 병렬로 계산된다. 허나 그림 4에서 와 같이 sequential decoding algorithm을 이용함으로써 동일한 성능에서 계산량의 감소라는 두 가지 이득을 얻을 수 있다^[8].

먼저 check node를 p개의 부분집합으로 나눈다. 각각의 부분집합은 n개의 check node를 가지고 있다. 이 때 총 check node의 수 $M = n \times p$ 이다. 그 다음 첫 번째 부분 집합의 check node를 업데이트 한다. 그 다음 업데이트된 check node를 통해서 bit node를 업데이트 하며 그 뒤 이때의 bit node를 통하여 다른 부분집합의 check node를 업데이트 하게 된다. 이때 부분집합을 나눌 때 각각의 check node들과 bit node들과의 edge를 광범위 하게 분포하는 것이 중요하다.

직렬로 연결된 구조에서 첫 번째 복호기가 복호를 한 후 외부정보(extrinsic information)를 두 번째 복호기에 전달되면서 복호 과정이 진행된다. 마찬가지로 sequential decoding 과정 또한 p개의 복호 모듈로 나뉘어 각각의 복호를 마친후(bit node, check node), 다음 복호 모듈로 전달된다. 이는 이미 bit node update 결과 후 다음의 부분집합으로 전달하기 때문에 기존의 방식에 비해 성능열화 없이 같은 복호 시간에 $N_r/2$ 만큼의 반복회수만 필요하다. (N_r : 기존 방식의 반복회수)

다음 그림 5는 부분집합 (p=2) 경우의 sequential decoder와 일반적인 decoder와의 BER 성능 곡선이



(A) 첫번째 부분 집합의 update (B) 두번째 부분 집합의 update

그림 4. Sequential 복호 과정

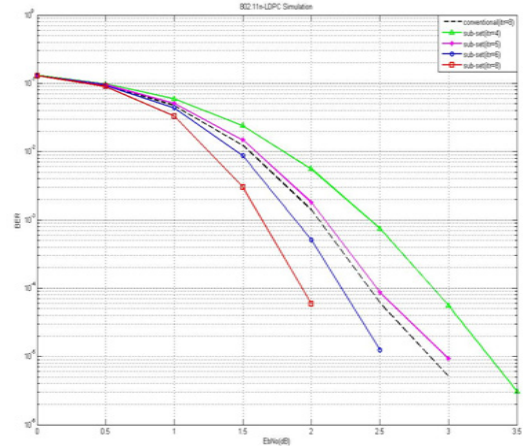


그림 5. sequential decoder와 기존 복호기의 성능비교

다. Sequential decoding algorithm 을 사용하였을 경우 2.2절의 부호화율 1/2에서 가우시안 잡음 채널 환경하에서 시뮬레이션 한 그림이다. 기존 방식의 8회의 반복회수와 sequential decoding algorithm 적용시 5회의 반복시 성능이 가장 근접함을 알 수 있다.

3.2 Early-stop algorithm

LDPC와 같은 반복 부호는 일정한 반복 후에는 더 이상 성능이 개선되지 않는 현상이 있으며, 이는 복호속도 저하 및 power소모량만 증가시키고는 결과만 맞고 있다. 따라서 적절한 early-stop 알고리즘을 이용하여 더 이상의 성능 개선이 없으면 반복을 중지시키는 알고리즘을 사용해야 한다.

본 논문에서 제시한 알고리즘은 각 iteration순간에 나온 LLR 값을 Hard decision 하여 비교하는 방식인 HDA(Hard Decision Aided Algorithm) 이다. 각 iteration에서 결정된 LLR을 HD 하여 그 다음 iteration과 비교, 더 이상의 변화가 없을 경우 decoding을 종료하고 최종적으로 나온 LLR을 이용하여 디코딩한다^[9].

표 2에서 알 수 있듯이 early stop 알고리즘을

표 2. E_b/N_0 에 따른 평균 반복 회수

E_b/N_0	Hard decision
	Iteration
0.5	8
1	8
1.5	7.9
2	7.8
2.5	6.9
3	5.8

적용시 E_b/N_0 가 3dB에서 약 3회 iteration을 적게 하여 계산량을 줄일수 있음을 알수 있다. 그림 6에서 보듯이 iteration 8회한 BER 성능과 위의 알고리즘을 적용하였을 때의 성능 곡선이 거의 일치함을 알 수 있다.

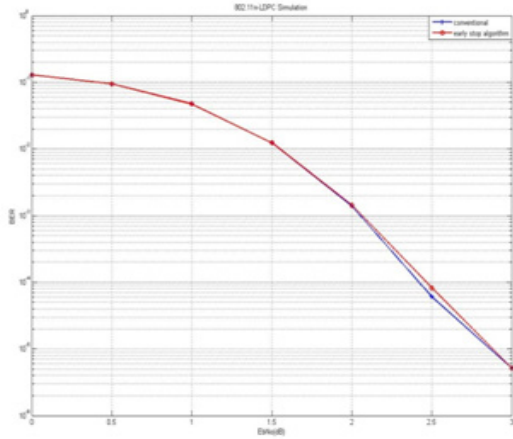


그림 6. Early stop algorithm의 BER 성능곡선(N=1944, R=1/2, irregular LDPC, 가우시안 잡음)

3.3 Early Detection Method

Decoding 과정의 복잡한 계산량을 줄이는 또 다른 방법은 Early Detection이다. 이 Early Detection 방식은 높은 log likelihood 값을 지닌 bit node나 check node는 신뢰할 수 있다는 개념을 기반으로 한다. 따라서 Detected 된 bit node나 check node는 다음 iteration에서 계산 과정을 거치지 않는다. bit node나 check node를 업데이트 할 때 일정한 기준 값을 넘게 되면 그 값은 더 이상 iteration이후의 업데이트를 하지 않더라도 신뢰할 만한 값으로 사용 가능 하게 된다.

Step1. initialization

모든 Detected Check node, bit node를 0 으로 초기화 시킨다.

Step2. Check node updates

Check node의 값이 일정한 기준 값(T_c) 이상 일 때 Detected check를 1로 설정하고 그렇지 않을 경우 check node update 공식에 따라 계산한다.

Step3. Bit node updates

Bit node의 값이 일정한 기준 값(T_b) 이상 일 때 Detected node를 1로 설정하고 그렇지 않을 경우 bit node update 공식에 따라 계산한다^[9].

그림 7에서 $T_c=9$ 로 고정시키고 T_b 값을 변화시키

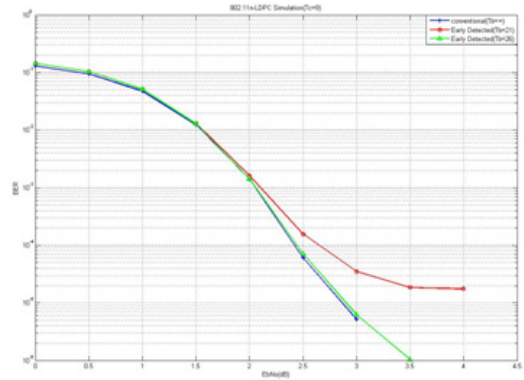


그림 7. $T_c=9$ 로 고정시 T_b 값 변화에 따른 BER 성능곡선 (N=1944,R=1/2,irregular LDPC, 가우시안 잡음)

면서 BER 성능을 나타낸 그림이다. $T_b=26$ 일 때 일반적인 LDPC decoder와 성능이 거의 일치함을 알 수 있다. 표 3은 LDPC decoder에서 요구되는 계산량의 나열한 것이다.

여기서 d_c^* 는 각 check node당 early detected 한 edge의 평균 숫자 이고, N^* 는 bit node를 early detected한 평균 숫자이다.

$T_c=9, T_b=26$ 으로 고정하였을 때 표 4는 E_b/N_0 가 에 따른 d_c^* 와 N^* 의 숫자를 보여준다. 이 때 $N=1944, K=972, M=972$, 이며 $d_c=8, d_v=11$ 이다. 표3에서 알 수 있듯이 각 check node당 8개의 edge가 있는데 $E_b/N_0=2.5$ 일 때 약 2.4개를 줄일 수 있으므로 check node update 계산시 약 30%이상의 계산량을 감소 할 수 있으며, 비트노드는

표 3. LDPC decoder에 요구된 계산량

	conventional decoder	early detection method
Check nodes(g(a,b))	$(M \times (d_c - 1)) \times N_r$	$(M \times (d_c - d_c^* - 1)) \times N_r$
Bit nodes (additions)	$(N \times (d_v - 1)) \times N_r$	$((N - N^*) \times d_v) \times N_r$

표 4. early detected 된 수

	E_b/N_0						
	0	0.5	1	1.5	2	2.5	3
d_c^*	0	0	0.037	0.21	1.396	2.401	3.429
N^*	293.074	390.973	675.71	1198.431	1671.479	1884.079	1934.663

$E_b/N_0=2.5$ 일 때 기존 방식은 1944이나 early detect한 비트노드는 약 1884이므로 약 97%이상의 계산량을 감소할 수 있다. 경관정과 비교하는 회로는 엣지를 구하는 체크노드와 비트노드의 확률값을 구하는 과정보다 상대적으로 계산량이 매우 적으므로 보통 논문에서는 계산량에 의미를 두지 않으므로 본 논문에서도 계산량에 고려를 하지 않았다.

IV. 결 론

반복 부호에서는 사논의 한계에 접근하기 위해서는 큰 블록 사이즈와 많은 반복횟수를 필요하며, 이는 많은 계산량을 요구하므로 성능 손실 없이 반복 횟수를 줄일 수 있는 알고리즘을 제시하였다. 반복 횟수를 기존의 방식에 비해 성능 열화 없이 줄이는 방법은 부분 병렬 방법을 이용한 복호 알고리즘, 그리고 early stop 알고리즘에 대해 연구 하였고, 그리고 각각의 반복시 계산량을 줄이기 위해, 비트 노드 계산과 체크 노드 계산시 일정한 신뢰도 값보다 크면 다음 반복시 계산을 하지 않는 early detection 알고리즘에 대해 연구 하였다. 802.11n 규격에 제시된 $N=1944$, $K=972$ $dc=8$, $dv=11$, $R=1/2$ 부호에서 시뮬레이션한 결과 부분 병렬을 이용하여 $p=2$ 일 때 성능열화가 거의 없이 반복회수를 거의 절반으로 줄일 수 있었으며, early stop 적용결과, $E_bN_0=3dB$ 에서 약 3회의 반복 횟수를 줄일 수 있다. 계산량을 줄일 수 있는 early detect 방식을 적용시킨 결과 check node 계산량은 약 30%, bit node계산량은 약 94%이상 감소함을 알 수 있다. 본 논문의 결과를 토대로 향후 802.11n 기반의 LDPC 복호 알고리즘에 본 논문에서 제시한 계산량 및 복잡도 감소 알고리즘을 적용가능하리라 사료된다.

참 고 문 헌

[1] "Satellite Broadcasting System of Integrated Service Digital Broadcasting", ITU-R BO.1227-2
 [2] "Digital Video Broadcasting(DVB): Framing Structure, Channel Coding and Modulation for Digital Satellite News Gathering(DSNG) and Other Contribution Applications by Satellite", ETSI EN 301 210 :European Standard
 [3] D. J. C. Mackay and R. M. Neal, "Near

Shannon Limit Performance of Low-Density Parity-Check Codes," Electron. Letter, Vol. 32, pp.1710-1722, Aug.1996.

[4] T.Richardson and R. Urbanke, "Efficient Encoding of Low-Density Parity-Check Codes," IEEE Trans. Informaion Theory, vol. 47, pp.638-656, Feb. 2001.
 [5] Brendan J. Frey and Frank R. Kschischang, "Early Detection and Trellis Splicing: Reduced-Complexity Iterative Decoding," IEEE Journal on Selected areas in communications, Vol.16, No.2, February 1998
 [6] 김성환외 2인, "체크노드 부분집합에 대한 순차적 복호를 이용한 LDPC 복호 방법", JCCI2004, 2004.
 [7] E.Eleftheriou, T.Mittelholzer and A.Dholakia, "Reduced complexity iterative decoding of low density parity check codes based on belief propagation," IEE Electron Letter, Vol.37, pp.102-104, Jan. 2001.
 [8] Z.Cai, J. Hao, P.H.Tan, S.Sum and P.S. Chin "Efficient encoding of IEEE 802.11n LDPC codes", IEEE Electronics Letters 7th December 2006 Vol.42, No.25.
 [9] 정지원, "DVB-S2 시스템을 위한 저복잡도 LDPC 복호 알고리즘," 한국전자과학회논문지, 16권 10호, 2005년 10월

김민혁 (Min-hyuk Kim)

준회원



2006년 2월 한국해양대학교
전파 공학과(공학사)
2008년 2월 한국해양대학교
전파 공학과(공학석사)
2008년 3월~현재 한국해양대
학교 전파공학과 박사과정

<관심분야> 위성 통신, 이동 통신, 변·복조 기술, 채널 코딩, FPGA 기술 등

박 태 두 (Tae-doo Park)

준회원



2008년 2월 한국해양대학교 전
파공학과 (공학사)
2008년 3월~현재 한국해양대학
교 전파공학과 석사과정
<관심분야> 위성통신, 이동통신,
변·복조기술, 채널코딩,
FPGA 기술 등

이 성 로 (Seong-ro Lee)

정회원



1987년 2월 고려대학교 전자
공학과
1990년 2월 한국과학기술원 전
기 및 전자공학과 석사
1996년 8월 한국과학기술원 전
기 및 전자공학과 박사
2005년 3월~현재 목포대학교
정보공학부 정보전자공학전공 부교수
<관심분야> 디지털통신시스템, 이동 및 위성통신시
스템, USN/텔레미틱스응용분야, 임베디드시스템,
생 체인식시스템

정 지 원 (Ji-won Jung)

정회원



1989년 2월 성균관대학교 전자
공학과(공학사)
1991년 2월 성균관대학교 전자
공학과(공학석사)
1995년 2월 성균관대학교 정보
공학과(공학박사)
1991년 1월~1992년 2월 LG
정보통신연구소 연구원
1995년 9월~1996년 8월 한국통신 위성통신연구실
선임연구원
1997년 3월~1998년12월 한국전자통신연구원 초빙
연구원
1996년 9월~현재 한국해양대학교 전파공학과 정교수
2001년 8월~2002년 8월 캐나다 NSERC Fellowship
(Communication Research Center 근무)
<관심분야> 위성통신, 이동통신, 변·복조기술, 채널
코딩, FPGA 기술 등

정 민 아 (Min-A Jung)

정회원



1992년 2월 전남대학교 전산통
계학과
1994년 2월 전남대학교 전산통
계학과 석사
2002년 2월 전남대학교 전산통
계학과 박사
2005년 3월~현재 목포대학교
컴퓨터공학과 조교수
<관심분야> 데이터베이스/데이터마이닝, 생체인식시
스템, 무선통신응용분야(RFID, USN, 텔레메틱스),
임베디드시스템