

5G를 위한 채널 코드 후보 기술과 연구 동향

박진수·김인선·  
김강산·송홍엽

연세대학교

I. 서론

현재의 4G 시대에 이르러, 스마트폰은 사용하지 않는 사람이 없을 정도가 되었고, 다양한 태블릿 PC, 스마트워치 등이 널리 보급되어 많은 수의 기기들이 무선 통신 네트워크에 참여하고 있다. 앞으로 더욱 많은 수의 단말들이 더욱 많은 양의 무선 데이터를 요구하게 될 것은 부정하기 어려워 보인다. 4G 기술이 확립된 이후, 폭발적으로 증가하는 데이터 요구량을 충족시키면서도 더 많은 기기에 더 고품질의 서비스를 제공할 수 있는 차세대 5G 기술을 개발하기 위해 다양한 연구들이 활발히 진행되어왔다. 이러한 5G 기술의 핵심 요구사항은 초고속(high data rate), 저지연(low latency), 저비용(ultra low cost), 고신뢰도(high reliability), 초다수 디바이스 연결(massive connectivity) 등이다<sup>[1]-[3]</sup>.

채널 코드는 통신 채널에서 발생할 수 있는 에러를 수정하기 위해 꼭 필요한 기술이다. 1990년대에 들어 Shannon limit에 근접한 성능을 보이는 turbo 코드가 발견된 이후<sup>[4]</sup>, low-density parity-check(LDPC) 코드<sup>[5]</sup>와 polar 코드<sup>[6]</sup> 등의 코드가 최근까지 활발하게 연구되고 있다. 최근 채널 코드의 연구 추세는 높은 오류 정정 능력을 가지면서도 빠른 en/decoding이 가능하도록 하여, 5G 통신 시스템의 초고속, 저지연, 저비용, 고신뢰도를 달성할 수 있게 하는 방향으로 이루어지고 있다.

본 고에서는 5G 시스템에 사용될 수 있는 채널 코드 후보인 turbo 코드, LDPC 코드, polar 코드와 연구 추세를 소개하고자 한다.

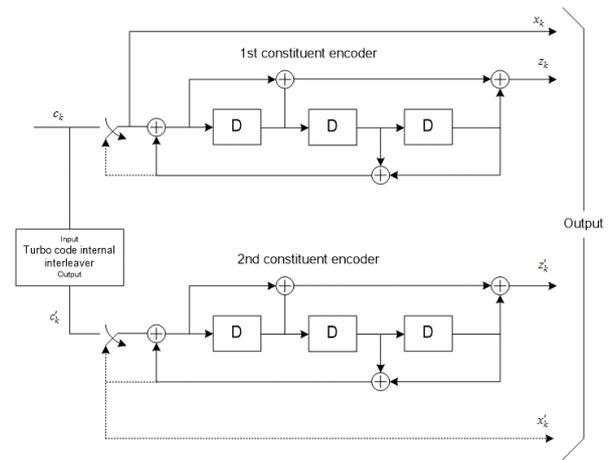
II. Turbo 코드

Turbo 코드는 현재 4G LTE 시스템을 비롯하여 가장 널리 사용되어온 코드 중 하나이다. Turbo 코드는 Shannon limit에 근접한 성능을 가지면서도 encoder 구조가 간단하며, 이미 널리 사용되어 충분히 성능과 동작이 검증된 기술이라는 장점이 있다.

2-1 Turbo Encoder

Turbo 코드의 encoder는 여러 개의 component encoder와 interleaver로 구성되어 있다. 본 고에서는 일반적으로 널리 사용되는 Parallel Concatenated Convolutional Code (PCCC)를 살펴해보도록 한다.

[그림 1]은 4G LTE 시스템에 사용되는 turbo encoder의 구조이다<sup>[7]</sup>. 각 convolutional encoder는 3개의 메모리를 갖는 linear feedback shift register(LFSR)로 구성되어 있으며, 첫 번째 convolutional encoder는 입력 메시지 bit  $c_k$ 가 들어올 때 패



[그림 1] 부호율 1/3의 turbo encoder 구조<sup>[7]</sup>

본 연구는 광주과학기술원 전자전통화연구센터를 통한 방위사업청과 국방과학연구소 연구비 지원으로 수행되었습니다.

리티 bit  $z_k$ 를 만들어내는 1/2 부호율의 코드이다. 두 번째 convolutional encoder는 동일한 구조를 갖지만, 입력 bit  $c_k$ 를 interleaving한  $c'_k$ 을 입력으로 사용하며, 이를 통해 만들어내는 패리티 bit  $z_k$ 만을 출력으로 내보낸다. 따라서 이 turbo 코드는 1 bit의 입력이 들어오면 해당 입력 1 bit와 패리티 2 bit를 출력하는 1/3 부호율의 코드이다.

Turbo 코드는 decoder의 동작성과 다음의 encoding을 위해 메시지 bit를 모두 입력한 후, encoder 내의 모든 메모리를 0이 되도록 하는 trellis termination을 수행하게 된다. 모든 입력 메시지 bit가 encoding된 후, [그림 1]의 LFSR 연결을 점선으로 표시된 것으로 바꾸어 각 convolutional encoder의 메모리 수인 3회만큼 encoding을 추가로 시행하면 모든 메모리가 0이 된다.

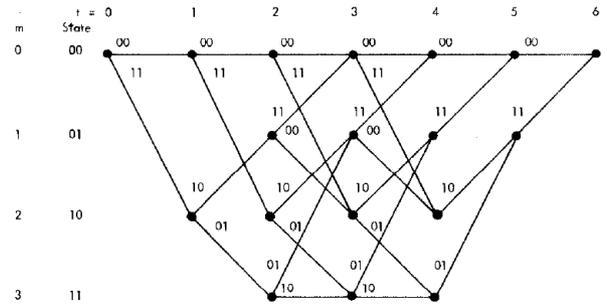
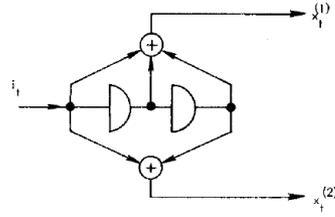
Turbo 코드를 표시하는 표준에는 일반적으로 convolutional encoder의 구조와 연결관계, 그리고 interleaver의 형태가 주어진다.

### 2-2 Turbo Decoder

Turbo 코드는 convolutional 코드로 구성되어 있으므로, convolutional decoder를 이용하여 decoding을 수행한다. Convolutional 코드의 decoding 기법은 여러 가지가 있으나, turbo code의 decoding을 위해서는 일반적으로 BCJR decoding이 사용된다<sup>[8]</sup>. BCJR decoding은 convolutional 코드의 maximum a posteriori(MAP) decoder임이 잘 알려져 있으며, turbo 코드의 확률적 반복 decoding을 위한 soft-input soft-output(SISO) decoding이 가능하다. 본 절에서는 log domain에서의 BCJR decoding을 소개한다.

한 convolutional 코드의 encoding은 LFSR을 이용한 Markov state transition 모델로 표현 가능하며, 이를 시간 순으로 표현하면 trellis 모델로 표현된다. [그림 2]는 메모리가 2개인 convolutional encoder와 그에 따른 trellis diagram을 보여준다. 하나의 codeword를 만들 때  $T$ 회의 state transition이 발생한다고 가정하자.

$x_t^{(1)}$ 과  $x_t^{(2)}$ 를 각각  $t$ 번째 입력  $i_t$ 에 해당하는 출력 bit라 하자. 수신기는 codeword를 수신한 후, 각  $x_t^{(1)}$ 과  $x_t^{(2)}$ 에 해당하는 log-likelihood ratio(LLR)  $L_{ch,t}^{(1)}$ ,  $L_{ch,t}^{(2)}$ 을 계산한다. 각 bit에 대한 a priori probability의 LLR을  $L_{a,t}^{(1)}$ ,  $L_{a,t}^{(2)}$ 라 할 때, 모든  $t$



[그림 2] 메모리가 2개인 convolutional encoder와 그에 따른 trellis diagram<sup>[8]</sup>

$= 0, \dots, T$ 에서 가능한 모든  $s' \rightarrow s$  state transition metric  $\gamma(s', s)$ 를 다음과 같이 계산한다.

$$\gamma(s', s) = (-1)^{\tilde{x}_t^{(1)}} \frac{L_{a,t}^{(1)} + L_{ch,t}^{(1)}}{2} + (-1)^{\tilde{x}_t^{(2)}} \frac{L_{a,t}^{(2)} + L_{ch,t}^{(2)}}{2}$$

이 때,  $\tilde{x}_t^{(1)}$ 과  $\tilde{x}_t^{(2)}$ 는  $s' \rightarrow s$  state transition에서 발생하는 각 출력 bit를 뜻한다. 이렇게 구해진  $\gamma$ 를 이용해 forward recursion을 통한  $\alpha$ , backward recursion을 통한  $\beta$ 를 구할 수 있다. 먼저 어떤 두 수  $x, y$ 에 대해  $x > y$ 일 때 다음과 같이 함수를 정의하자.

$$\max^*(x, y) = x + \log((1 + e^{-x}))$$

$x \leq y$ 일 때에는  $\max^*(y, x)$ 로 계산한다. 변수가 여러 개일 경우, 2개의 변수씩 차례대로 계산한다.  $t = 0$ 일 때,  $s \neq 0$ 인 state에 대해  $\alpha_0(s) = -\infty$ ,  $s = 0$ 인 state에 대해  $\alpha_0(0) = 0$ 으로 초기화 한다. 그리고 다음과 같은 forward recursion을 수행한다.

$$\alpha_t(s) = \max_{s'}^* (\alpha_{t-1}(s') + \gamma(s', s))$$

마찬가지로,  $t = T$ 일 때  $s \neq 0$ 인 state에 대해  $\beta_T(s) = -\infty$ ,  $s = 0$ 에 대해  $\beta_T(s) = 0$ 로 초기화 한 후 다음과 같은 backward recursion을 수행한다.

$$\beta_{t-1}(s') = \max_s^* (\beta_t(s) + \gamma_t(s', s))$$

모든  $a$ 와  $\beta$ 를 구한 후, 출력으로 a posteriori probability의 LLR  $L_{p,t}^{(1)} = L_{a,t}^{(1)} + L_{ch,t}^{(1)} + L_{c,t}^{(1)}$ ,  $L_{p,t}^{(2)} = L_{a,t}^{(2)} + L_{ch,t}^{(2)} + L_{c,t}^{(2)}$ 를 생성한다. 이 때,

$$L_{c,t}^{(1)} = \max_{(s',s)_0^{(1)}}^* (\alpha_{t-1}(s') + (-1)^{\tilde{x}_t^{(2)}} \frac{L_{a,t}^{(2)} + L_{ch,t}^{(2)}}{2} + \beta_k(s)) - \max_{(s',s)_1^{(1)}}^* (\alpha_{t-1}(s') + (-1)^{\tilde{x}_t^{(2)}} \frac{L_{a,t}^{(2)} + L_{ch,t}^{(2)}}{2} + \beta_k(s))$$

$$L_{c,t}^{(2)} = \max_{(s',s)_0^{(2)}}^* (\alpha_{t-1}(s') + (-1)^{\tilde{x}_t^{(1)}} \frac{L_{a,t}^{(1)} + L_{ch,t}^{(1)}}{2} + \beta_k(s)) - \max_{(s',s)_1^{(2)}}^* (\alpha_{t-1}(s') + (-1)^{\tilde{x}_t^{(1)}} \frac{L_{a,t}^{(1)} + L_{ch,t}^{(1)}}{2} + \beta_k(s))$$

이다.  $(s', s)_x^{(y)}$ ,  $x = 0, 1$ ,  $y = 0, 1$ 는  $\tilde{x}^{(y)} = x$ 인  $s' \rightarrow s$  state transition을 의미한다.

Turbo decoder는 두 개의 convolutional encoder에 따른 두 개의 BCJR decoder를 갖는다. 각 BCJR decoder는 각각 독립적으로 decoding을 수행하여 출력 LLR  $L_p$ 를 만들어낸다. 첫 번째 decoder의 출력은 interleaver를 사용하여 두 번째 decoder의 a priori  $L_a$ 로 사용한다. 마찬가지로, 두 번째 decoder의 출력은 deinterleaver를 사용하여 첫 번째 decoder의 a priori  $L_a$ 로 사용한다. 이러한 과정을 수회 반복한 후, 첫 번째 decoder의 출력 LLR  $L_p$ 를 이용해 메시지 bit를 추정한다.

### III. LDPC 코드

LDPC 코드는 4G WiMax와 Wi-Fi, 방송 등의 시스템에 사용되고 있다. LDPC 코드는 decoder 구조가 병렬 처리가 가능하여 빠른 복호가 가능하다. 또한, Shannon limit에 근접한 매우 높은 성능을 보인다. LDPC 코드 역시 널리 사용되어 성능이 검증되었으며, 현재에도 고성능 LDPC 코드에 대한 연

구가 꾸준히 이어지고 있다.

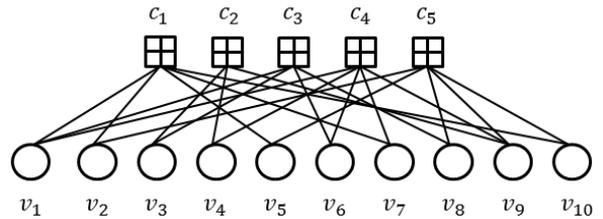
#### 3-1 LDPC Decoder

LDPC 코드는 turbo 코드와는 다르게 decoder(패리티 검사 행렬 H)의 구조를 이용해 정의된다<sup>[9]</sup>. <표 1>은 길이  $N=10$ , 메시지 길이  $K=5$ 에 해당하는  $(N-K) \times N$  패리티 검사 행렬의 예시이다. 각 행은 하나의 check node(CN,  $c_i$ ,  $i = 1, \dots, 5$ ), 각 열은 하나의 variable node(VN,  $v_j$ ,  $j = 1, \dots, 10$ )을 나타낸다. 이들의 관계는 [그림 4]와 같은 이분 그래프(bipartite graph)로 표현되며, 일반적인 LDPC 코드의 분석과 설계는 그래프 상에서 이루어진다. [그림 4]의 田는 CN, ○는 VN을 의미한다. 패리티 검사 행렬의 각 행은 하나의 패리티 검사 방정식을 의미하며, 해당 방정식에 참여하는 변수(VN)의 합은 0이 되도록 encoding 된다. 따라서 [그림 3]의 첫 번째 CN  $c_1$ 은 H의 첫 번째 행에 총 5개의 1을 가지고 있으며, 이는 각각 1, 2, 5, 7, 10번째 열에 위치하고 있다. 즉, 1, 2, 5, 7, 10번째 code-word bit를 binary field에서 더하면 0이 되어야 하며, 또한 [그림 4]에서  $c_1$ 에 연결된  $v_1, v_2, v_5, v_7, v_{10}$ 에 해당하는 bit들의 합이 0이 되어야 함을 뜻한다. 이것은 다른 CN에 대해서도 동일하게 표현이 가능하다.

이러한 LDPC 코드는 일반적으로 belief propagation(BP)

$$H = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

[그림 3]  $N = 10$ ,  $K = 5$  패리티 검사 행렬의 예



[그림 4] 그림 3의 패리티 검사 행렬에 해당하는 이분 그래프 (bipartite graph)



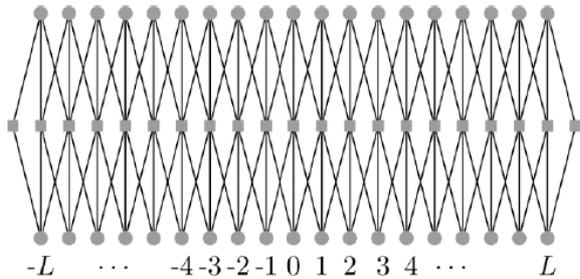
tional encoder를 사용한 LDPC 코드 설계 기법인 LDPC convolutional 코드, protograph 기반의 LDPC 코드의 연장선상에 있는 코드이다. [그림 6]은 protograph 기반의 spatial coupling 예시를 보여준다. 기존 SC-LDPC 코드의 연구는 이진 소거 채널(binary erasure channel; BEC)에서 주로 연구되었으나, BEC와 AWGN 채널 모두 좋은 성능을 갖는다는 것이 알려졌다<sup>[24],[25]</sup>.

#### IV. Polar 코드

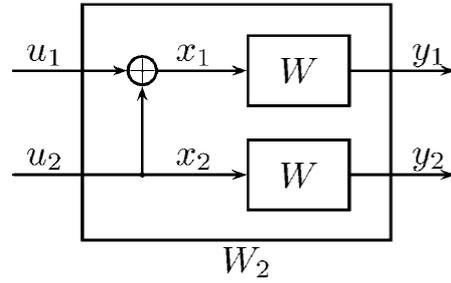
Polar 코드는 2008년 Arikan이 최초로 제안한 채널 양극화 (polarization) 현상을 이용하는 코드이다<sup>[6]</sup>. Polar 코드는 turbo 코드와 LDPC 코드를 대체할 수 있는 후보 기술로써 활발히 연구되고 있다.

##### 4-1 채널 양극화와 Polar Encoder

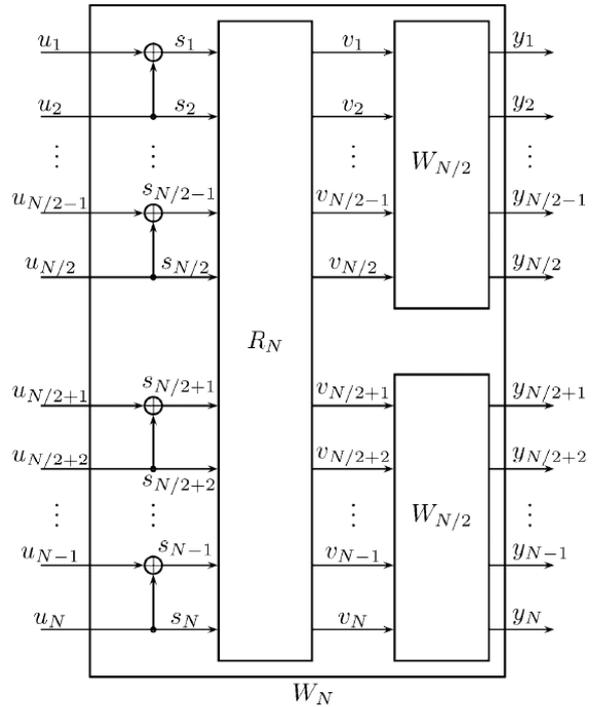
채널 양극화는 두 개의 동일한 용량의 채널이 서로 다른 용량(capacity)으로 변하는 현상이다. [그림 7]은  $W$ 라는 동일한 용량  $I(W)$ 의 두 채널이 서로 다른 용량의 두 채널로 양극화 시키는  $N = 2$  채널 결합  $W_2$ 를 보여준다<sup>[6]</sup>. 입력 bit  $u_1$ 과  $u_2$ 는 각각  $x_1 = u_1 + u_2$ ,  $x_2 = u_2$ 로 보내지며  $x_1$ 과  $x_2$ 는  $I(W)$  용량의 채널을 통해 보내진다. 이 때,  $u_1$ 과  $u_2$ 가 느끼는 채널 용량  $I(W_2^{(1)})$ ,  $I(W_2^{(2)})$ 는  $I(W_2^{(1)}) = I(W)^2$ ,  $I(W_2^{(2)}) = 2I(W) - I(W)^2$ 로 양극화 된다. 즉,  $u_1$ 과  $u_2$ 는 서로 다른 채널 용량을 사용하는 것과 같은 상태가 된다. Polar 코드는 이러한 용량 양극화를 반복적으로 사용하며, 이는 [그림 8]과 같이 표현된다.  $W_{N/2}$  두 개와 길이  $N$ 의  $R_N$ 을 통한 shuffle을 이용하면  $W_N$ 을 만들 수 있다. 이를 반복하면 [그림 9]와 같이 채널 용량이 양



[그림 6] SC-LDPC 코드의 protograph coupling<sup>[22]</sup>



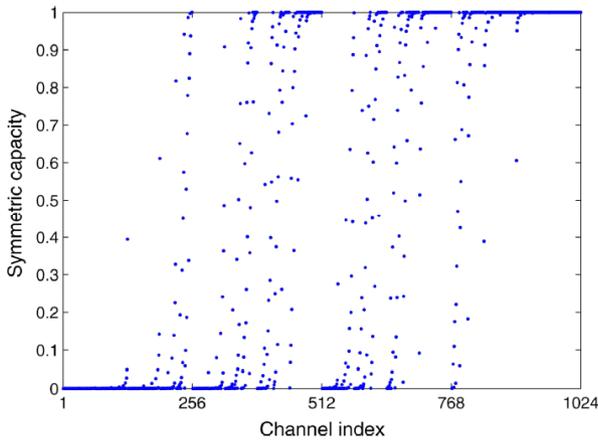
[그림 7]  $W_2$  채널 결합



[그림 8]  $W_{N/2}$ 로부터 얻어지는  $W_N$ <sup>[6]</sup>

극화 되는 것을 볼 수 있다. 대체적으로 낮은 index의 채널들이 0의 용량을 가지며, 높은 index의 채널들이 1의 용량을 갖는다. 일부 0과 1사이의 용량을 갖는 채널들이 존재한다.

일반적으로 polar 코드는 이렇게 구해진  $N$ 개의 입력 bit  $u_i$ 에 해당하는 채널 용량을 구하고, 이 중 가장 채널 용량이 좋은  $K$ 개의 index를 선택하는 것으로 설계된다. 선택되지 않은  $N-K$ 개의 bit는 사전에 결정된 bit(일반적으로 0)가 입력되며, 이들을 frozen bit라 부른다. 이렇게  $K$ 개의 index에 메시지 bit를 입력하고,  $N$  bit의 출력을 codeword라 하면 곧  $(N -$

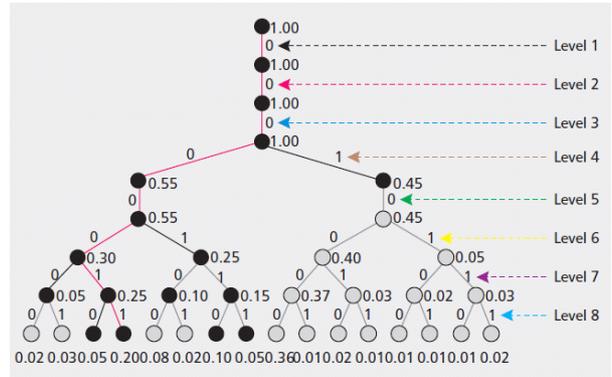


[그림 9] 소실율 50%인 BEC에서  $N = 2^{10}$ 일 때  $i = 1, \dots, N$ 에 대한  $I(W_N^{(i)})^{[6]}$

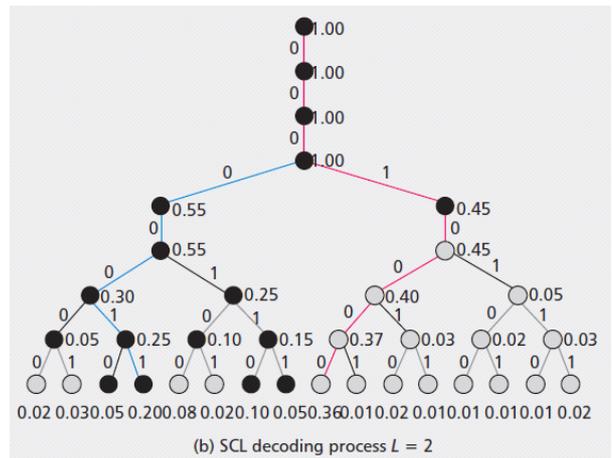
$K/N$  부호율의 polar 코드가 된다.

#### 4-2 Polar Decoder와 최근 연구 동향

Polar 코드의 기본적인 decoding 기법은 successive cancellation(SC) decoding이다. 이는 메시지 index  $i = 1$ 부터 차례대로  $i = N$ 까지 차례대로 메시지 입력  $u_i$ 를 찾아내는 기법이다. Polar 코드의 decoding 과정은 [그림 10], [그림 11]과 같이 tree로 표현된다<sup>[26]</sup>. [그림 10]과 [그림 11]에서는  $N = 8$ 이며,  $i = 1, 2, 3, 5$ 의  $u_i$ 들이 0으로 frozen 되었다고 가정하자. 각  $i$ 번째 level의 ●와 ○들은  $u_i$ 에 해당하며, decoding 과정 중에 process된 노드는 ●, 그렇지 않으면 ○ 표기되었다. 각 노드에 적힌 수는 root 노드부터 해당 노드까지 bit 수열에 대한 확률(path metric)이며 각 엣지에 적힌 0과 1은  $i$ 번째 bit가 0일 경우와 1일 경우를 나타낸다. [그림 10]을 보면, 가장 처음 3개의 bit는 frozen bit이므로 1.00의 확률로  $(u_1, u_2, u_3) = (000)$ 이 된다. 4번째 bit(Level 4)에 도달하면  $u_4$ 가 0일 확률 1일 확률이 각각 0.55, 0.45이므로 0이 선택된다. Level 4에서  $(u_1, u_2, u_3, u_4) = (0000)$ 과  $(0001)$ 에 대한 metric을 계산했으므로 Level 4의 두 노드 모두 ● 표시되었다. 다음 Level 5는 frozen bit이므로  $u_5 = 0$ 이 고정되고 Level 6으로 넘어간다. Level 6에서는  $(u_1, u_2, u_3, u_4, u_5, u_6) = (000000)$ 과  $(000001)$ 의 확률이 각각 0.30, 0.25이다. 따라서  $(u_1, u_2, u_3, u_4, u_5, u_6) = (000000)$ 이 선택되고, Level 7로 넘어간다. 이러한 과정을 Level 8까지 거



[그림 10] SC decoding의 예제<sup>[26]</sup>



[그림 11] SCL decoding ( $L = 2$ )의 예제<sup>[26]</sup>

치면, 최종적으로 (0000011)이 선택된다.

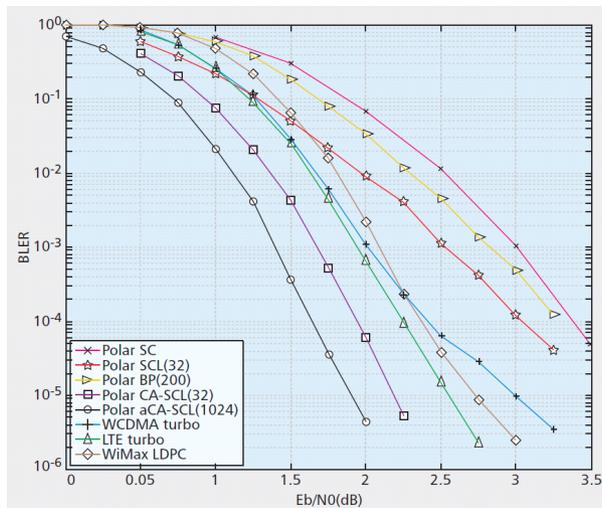
이러한 SC decoding 알고리즘은 실제 사용 환경에서는 좋은 성능을 보이지 못하는 단점이 있다. Polar 코드의 복호 성능을 향상시키기 위해 SC decoding 기법에 list와 stack을 도입한 SCL decoding과 SCS decoding이 제안되었다<sup>[27],[28]</sup>. 본 고에서는 [그림 11]의 예제를 통해 SCL decoding을 살펴보도록 하자. 예제에서는 한번에 2개의 path(list)를 다루는  $L = 2$ 의 경우를 보여준다. 우선 Level 1, 2, 3은 frozen bit이므로 (000)이 동일하게 구해진다. 이후, 4번째 Level에서 0과 1일 경우로 나뉘는데, 2개까지의 path를 다루기로 했으므로 (0000)과 (0001)일 때의 path metric 0.55, 0.45를 각각 저장한다. 그리고 Level 5의 frozen bit를 지나 Level 6에서 (00000)과 (00010)에 대해  $u_6 = 0, 1$ 일 때의 path metric을 조사한다. [그림 11]

에는 각각 0.30, 0.25, 0.40, 0.05로 나타나 있다. 우리는  $L = 2$  개의 path만을 조사하고 있으므로, 가장 큰  $L$ 개의 path metric에 해당하는 (000000)과 (000100)만을 저장하고, Level 7로 넘어간다. 마찬가지로  $u_7 = 0, 1$ 일 때의 path metric 4개를 계산하여 가장 큰 값을 갖는 2개의 path를 저장한다. 이런 과정을 거쳐 최종적으로  $L$ 개의 path 중 가장 높은 path metric을 갖는 path를 선택하는 것으로 decoding을 종료한다.

이러한 SCL decoding 기법의 성능을 더욱 향상시키기 위해 cyclic redundancy check(CRC)와 연결한 CRC-aided SCL(CA-SCL) decoding 기법이 제안되었다<sup>[29]</sup>. CA-SCL decoding은 SCL decoding을 이용해 찾은 path에 대해 CRC 검사를 하여 이를 통과한 path를 선택한다.

### V. 결 론

[그림 12]는 WCDMA와 LTE의 turbo 코드, WiMax의 LDPC 코드, 그리고 Polar 코드의 다양한 decoding 기법에 따른 BLER을 나타낸 것이다<sup>[26]</sup>. 코드 길이는 1024(LDPC 코드의 경우 1056)로 동일하며, 부호율은 1/2이다. 우선 LTE turbo 코드와 WiMax LDPC 코드를 비교하면, LTE turbo 코드가 조금 더 좋은 성능을 보이는 것을 확인할 수 있다. Polar 코드의 SC decoder는 기존 turbo, LDPC 코드보다 많이 열화된 성능을 보이지만, CA-SCL decoding을 적용하면 LTE turbo 코드



[그림 12] Turbo 코드와 LDPC 코드, 다양한 polar 코드의 성능<sup>[26]</sup>

보다 더 좋은 성능을 보이는 것을 확인할 수 있다.

본 고에서는 5G 시스템을 위한 채널 코드의 후보로 거론되는 turbo, LDPC, polar 코드에 대하여 살펴보았다. Turbo 코드는 좋은 성능과 간단한 encoder 구조를 가지지만, decoder의 병렬화가 어려운 문제점을 가진다. 하지만, 오랜 기간 다양한 시스템에 사용되어 검증되어 왔고, 기존 시스템과 호환이 가능하다는 장점이 있다. LDPC 코드는 높은 decoder 병렬화가 가능하며, 다양한 시스템에 적용되어 검증되었다는 장점이 있다. 최근, 기존 표준에 사용된 LDPC 코드보다 더 좋은 성능을 갖는 LDPC 코드들이 많이 연구되고 있다. 끝으로 polar 코드는 새로이 등장한 고성능의 코드로 높은 성능을 가진다는 장점이 있지만, 기존 시스템과 호환이 어렵다는 단점이 있다. 이러한 코드들은 5G를 위한 초고속, 저지연, 고신뢰도를 달성하는 방향으로 연구되고 있다.

### 참 고 문 헌

- [1] Nokia, "5G use cases and requirements", White Paper, 2014.
- [2] <http://www.huawei.com/5gwhitepaper/>
- [3] 박옥선, 황현용, 이창희, 신재승, "5G Massive IoT 기술 및 표준화 동향", 전자통신동향분석, 31(1), pp. 68- 77, 2016.
- [4] C. Berrou, A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes", *IEEE Transactions on Communications*, vol. 44, no. 10, Oct. 1996.
- [5] D. J. C. MacKay, R. M. Neal, "Near shannon limit performance of low density parity check codes", *Electronics Letters*, vol. 33, no. 6, Mar. 1997.
- [6] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binar-input memoryless channels", *IEEE Transactions on Information Theory*, vol. 55, no. 7, Jul. 2009.
- [7] [http://www.3gpp.org/ftp/Specs/2011-03/Rel-10/36\\_series/36212-a10.zip](http://www.3gpp.org/ftp/Specs/2011-03/Rel-10/36_series/36212-a10.zip)
- [8] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate", *IEEE Transactions on Information Theory*, Mar. 1974.
- [9] R. G. Gallager, "Low-density parity-check codes", *IRE Transactions on Information Theory*, Mar. 1963.

- sactions on Information Theory*, vol. 8, issue 1, Jan. 1962.
- [10] S. J. Johnson, *Iterative Error Correction*, Cambridge University Press, 2010.
- [11] <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1603394>
- [12] 김준성, 배슬기, 정비용, 송홍엽, "IEEE 802.16e의 LDPC 부호화 기술 분석", *텔레콤*, 20(2), pp. 27-33, 2004년 12월.
- [13] Z. Cai, J. Hao, P. H. Tan, S. Sun, and P. S. Chin, "Efficient encoding of IEEE 802.11n LDPC codes", *Electronics Letters*, vol. 42, no. 25, Dec. 2006.
- [14] M. C. Davey, D. MacKay, "Low-density parity check codes over GF(q)", *IEEE Communications Letters*, vol. 2, no. 6, Jun. 1998.
- [15] D. Declercq, M. Fossorier, "Decoding algorithms for non-binary LDPC codes over GF(q)", *IEEE Transactions on Communications*, vol. 55, no. 4, Apr. 2007.
- [16] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low-complexity decoding for non-binary LDPC codes in high order fields", *IEEE Transactions on Communications*, vol. 58, no. 5, May 2010.
- [17] X. Chen, C.-L. Wang, "High-throughput efficient nonbinary LDPC decoder based on simplified min-sum algorithm", *IEEE Transactions on Circuits and Systems-I: Regular Papers*, vol. 59, no. 11, Nov. 2012.
- [18] C.-L. Wang, X. Chen, Z. Li, and S. Yang, "A simplified min-sum decoding algorithm for non-binary LDPC codes", *IEEE Transactions on Communications*, vol. 61, no. 1, Jan. 2013.
- [19] E. Li, D. Declercq, and K. Gunnam, "Trellis-based extended min-sum algorithm for non-binary LDPC codes and its hardware structure", *IEEE Transactions on Communications*, vol. 61, no. 7, Jul. 2013.
- [20] J. O. Lacruz, F. G.-Herrero, J. Valls, and D. Declercq, "One minimum only trellis decoder for non-binary low-density parity-check codes", *IEEE Transactions on Circuits and Systems-I: Regular Papers*, vol. 62, no. 1, Jan. 2015.
- [21] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes", *IEEE Transactions on Information Theory*, vol. 56, no. 10, Oct. 2010.
- [22] S. Kudekar, T. J. Richardson, and R. L. Rubanek, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC", *IEEE Transactions on Information Theory*, vol. 57, no. 2, Feb. 2011.
- [23] D. J. Costello, Jr., L. Dolecek, T. E. Fuja, J. Kliewer, D. G. M. Mitchell, and R. Smarandache, "Spatially coupled sparse codes on graphs: Theory and practice", *IEEE Communications Magazine*, Jul. 2014.
- [24] L. Schmale, K. Mahdaviani, "Laterally connected spatially coupled code chains for transmission over unstable parallel channels", in *Proc. of 8th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, 2014.
- [25] Md. N.-A-Rahim, K. D. Nguyen, and G. Lechner, "Anytime reliability of spatially coupled codes", *IEEE Transactions on Communications*, vol. 63, no. 4, Apr. 2015.
- [26] K. Niu, K. Chen, J. Lin, and Q. T. Zhang, "Polar codes: Primary concepts and practical decoding algorithms", *IEEE Communications Magazine*, Jul. 2014.
- [27] I. Tai, A. Vardy, "List decoding of polar codes", *IEEE Transactions on Information Theory*, vol. 61, no. 5, May 2015.
- [28] K. Niu, K. Chen, "Stack decoding of polar codes", *Electronics Letters*, vol. 48, no. 12, 2012.
- [29] K. Niu, K. Chen, "CRC-aided decoding of polar codes", *IEEE Communications Letters*, vol. 16, no. 10, Oct. 2012.

≡ 필자소개 ≡

박진수



2009년 2월: 연세대학교 전기전자공학과 (공학사)  
2009년 3월~현재: 연세대학교 전기전자공학과  
석박사 통합과정  
[주 관심분야] 통신공학, 부호이론

김강산



2016년 2월: 연세대학교 전기전자공학과 (공학사)  
2016년 3월~현재: 연세대학교 전기전자공학과  
석박사 통합과정  
[주 관심분야] 통신공학, 부호이론

김인선



2012년 8월: 연세대학교 전기전자공학과 (공학사)  
2012년 9월~현재: 연세대학교 전기전자공학과  
석박사 통합과정  
[주 관심분야] 통신공학, 부호이론

송홍엽



1984년 2월: 연세대학교 전자공학과 (공학사)  
1986년 5월: University of Southern California Dept.  
of EE. Systems (공학석사)  
1991년 12월: University of Southern California  
Dept. of EE. Systems (공학박사)  
1992년 1월~1993년 12월: Post-Doc Research Asso-  
ciate, University of Southern California Dept.  
of EE. Systems  
1994년 1월~1995년 8월: Senior Engineer, Qualcomm Inc., San Diego,  
California.  
2002년 3월~2003년 2월: Visiting Professor, University of Waterloo,  
Canada  
1995년 9월~현재 : 연세대학교 전기전자공학과 교수  
[주 관심분야] 통신공학, 정보이론, 부호이론, 암호이론, 이산수학