JANT 한국항행학회논문지
Journal of Advanced Navigation Technology

응용 및 융합 기술

J. Adv. Navig. Technol. 19(2): 153-160, Apr. 2015

Unicode 한글낱자의 UTF-8 부호화에 따른 HDB-3 스크램블링 발생빈도

Scrambling Occurrence Frequency in HDB-3 in UTF-8 Coding of UNICODE Hangul Jamo

홍 완 표
한세대학교 정보통신공학과

Wan-pyo Hong
Department of Information and Telecommunication Engineering, Hansei University, Gyeonggi-Do 435-742, Korea

[요 약]

본 논문은 국제적 문자부호체계인 유니코드 내에 있는 한글낱자와 호환용 한글낱자 부호를 통신망에 전송하기 위해 UTF-8부호로 변환할 때 회선부호기에서 발생하는 스크램블링의 발생빈도를 연구하였다. 본 논문에서 적용한 회선부호기의 스크램블링 방식은 ITU 및 한국의 표준전송방식인 HDB-3방식으로 하였다. 각 한글낱자부호에서 발생하는 스크램블링을 분석하기 위해 원천부호화규칙을 적용하였다. 스크램블링 발생량은 각 한글낱자에서 발생하는 스크램블링의 발생횟수와 각 한글낱자의 사용빈도에 의한 발생 빈도율을 추출하였다. 연구결과 한글낱자의 부호에 대한 스크램블링 발생은 유니코드 체계내에서 24번, 52%, UTF-8체계 내에서 148번, 228%발생하였다. 호환용 한글낱자 부호에서는 유니코드 체계내에서 10번, 14%, UTF-8 체계 내에서 83번, 131% 발생하였다. 즉, 유니코드체계의 한글낱자와 호환용 한글낱자는 UTF-8체계로 변환하면서 스크램블링 발생 빈도율이 각각 340%, 851%증가하는 것으로 나타났다.

[Abstract]

This paper has studied about the scrambling occurrence frequency in UTF-8 coding system for Unicode Hangul Jamo codes. The scrambling methode applied in the study is HDB-3 in AMI line coding that is international transmission standard. In the study, the source coding rule was applied to analysis the scrambling occurrence. The quantity of the scrambling occurrence was calculated by the number of times and frequency rate of the scrambling occurrence in Hangul Jamo and Compatibity Hangul Jamo. In the case of Hangul Jamo, the number of times and frequency rate in Unicode and UTF-8 were 24times, 52% and 148times, 228% respectively. In the case of Compatibility Hangul Jamo, that were 10times, 14% and 83times, 131% respectively. As a result, when Hangul Jamo and Compatibility Hangul Jamo in UNICODE were transformed to UTF-8, the scrambling frequency rates were increased 340% and 851% respectively.

Key word : Unicode, Universal code system transformation format (UTF)-8, Source coding, Line coding, Scrambling.

http://dx.doi.org/10.12673/jant.2015.19.2.153

Received 15 March 2015; **Revised** 6 April 2015
Accepted (Publication) 10 April 2015 (30 April 2015)

***Corresponding Author; Wan-pyo Hong**

Tel: +82-31-45-5340
E-mail: wphong@hansei.ac.kr

Ⅰ. 서 론

정보기기에서 생성되는 문자의 원천부호체계는 생성된 정보를 통신망에 전송할 때 전송효율에 영향을 미친다. 정보기기에 입력되는 문자를 정해진 비트체계로 표현하는 것을 원천부호화라고 한다. 이 원천부호화는 OSI계층의 6계층, 즉 표현계층에서 실현된다. 이렇게 정보기기에서 생성된 원천문자부호는 물리계층에서 회선부호화 과정을 거쳐 통신망에 적합한 신호로 변환된다[1]. 원천문자부호는 회선부호화과정을 거쳐 그대로 통신망에 전송되거나 통신망에 적합한 포맷으로 변환되어 회선부호화된다. Unicode의 경우에는 원천부호가 UTF(universal code system transformation format)부호체계로 변환된다. 회선부호화방식은 RZ(return to zero), NRZ(non return to zero), Manchester, AMI(altenate mark inversion)등이 있다. 본 논문에서는 장거리 통신용 회선부호화방식인 AMI 방식을 연구대상으로 하였다. AMI방식은 원천부호가 회선부호기에 입력될 때 입력되는 부호중에 있는 +성분의 비트(일반적으로 비트 1)를 입력되는 순서대로 +와 -신호로 교대로 생성되게 한다. 이렇게 하므로서 전송로상에서의 누적 신호 평균 전력값이 0이 되게 한다. 또한 신호값의 극성이 수시로 바뀌는 것을 이용하여 비트의 동기를 맞추게 된다. 그런데 이 AMI방식의 경우, 원천부호의 +(비트 1) 성분만을 사용하여 회선부호화하기 때문에 회선부호기에 일정갯수 이상의 연속된 0비트로 구성된 원천부호가 입력될 경우, 회선부호기의 출력신호에 0전압이 연속되게 되어, 수신기에서 비트경계를 구분할 수 없게 되는 문제를 갖고 있다. 이 문제를 해결하기 위해 AMI방식은 스크램블링이라는 기능을 사용하고 있다. 즉 원천부호로부터 일정갯수 이상의 0비트가 연속해서 입력될때는 그 연속 0비트열을 인위적으로 0과 1이 반복되는 형태의 신호체계로 변환시켜서 출력시킨다. 즉 표현계층에서 원천부호가 어떠한 비트열의 형태로 부호화되어 있는가에 의하여 물리계층의 회선부호기의 효율에 영향을 주게 된다. AMI방식에서 사용하는 스크램블링방식에는 대표적으로 HDB-3방식과 B8ZS방식이 있다[2],[3]. 본 논문에서는 국내외 표준방식인 HDB-3방식을 적용하여 연구하였다. Unicode에는 영어, 한국어, 중국어, 일본어 등 전 세계의 문자가 부호화되어 있다. 한국어의 경우에는 한글 11,161자외에 한글자모들이 부호화되어 있다. 본 논문에서는 한글자모와 호환용 한글 자모를 연구대상으로 하였다. 이 연구대상 한글자모부호에 대한 스크램블링의 빈도를 분석하기 위해 2005년도 국립국어원에서 연구한 한국어 자모통계를 사용하였다[4]. 한글자모부호중에서 스크램블링이 발생하는 자모부호를 분석하기 위해 문자의 원천부호화 규칙을 적용하였다[5].

Ⅱ. Unicode와 UTF-8부호체계

2-1 유니코드(Unicode) 부호 체계

유니코드는 기본 2바이트 16비트체계로 시작되었다. 따라서 총65,536개의 문자를 부호화할 수 있다. 현재의 유니코드 체계는 3.0 버전으로 4바이트 32비트 체계이다. 이 4바이트 부호 체계는 상위, 하위 각각 2바이트 부호체계로 구성되어 있다. 상위부호는 문자판(plane)을 식별하기 위한 것으로 식별가능 문자판은 65,536개이다. 하위부호는 각 상위문자판에 대한 문자와 기호에 대한 부호를 갖는다. 즉 각 문자판에는 총 65,536개의 문자와 기호에 대한 부호를 부여하게 된다. 그러므로 유니코드 버전3.0은 총 4,294,967,296개의 문자와 기호에 대한 부호를 부여할 수 있다[12]. 표 1에서 BMP(basic multilingual plane)는 기본 다국어 문자판이다. 상위 2바이트는 16진수 “0000”이다. SMP(supplementary multilingual plane)는 추가 다국어 문자판으로 BMP에 포함되지 않은 다국어 문자부호를 갖는다. SIP(supplementary ideographic plane)는 한자 등 상형문자, 기호 그리고 발음과 대조를 이루는 의미를 표시하기 위한 기호에 대한 부호체계이다. SSP(supplementary special plane)는 특수문자 부호체계이다. PUA(private use area)는 사용자가 개인적으로 부호를 부여하도록 한 체계이다[6]. 유니코드 2001년 버전3.0에 부여된 문자수는 1991년 버전1.0에 부여된 문자수보다 13배 증가하였다[7].반면에 2001년 버전3.0과 2012년 버전 6.1간에는 1.6배의 증가 차이만 있다.

2-2 UTF-8 부호체계

UTF-8 부호체계는 1(8비트)~3바이트(24비트)의 3가지 부호체계를 가지고 있다[8],[5],[10]. UTF-8 부호체계는 최상위 숫자가 16진수 E로 시작된다. 표 1은 이 부호체계에 대한 예를 보여주는 것이다. 여기에서 1110, 10과 10의 비트는 유니코드를 UTF-8부호로 만들기 위해 정해진 비트들이다. aaaa는 유니코드의 최상위 4비트이고 bbbb는 두 번째 비트열, cccc는 세 번째 비트열 그리고 dddd는 최하위 비트열이다.

표 1. 유니코드의 UTF-8 부호 변환 예
Table 1. Example of UTF-8 encoding of unicode.

UTF-8 Transformation Rule	1	1	1	0	A	A	A	A	1	0	B	B	B	B	C	C	1	0	C	C	D	D	D	D
Unicode Binary (1100)	1	1	1	0	0	0	0	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0

Ⅲ. 원천부호화규칙과 유니코드 및 UTF-8 부호체계

3-1 유니코드와 원천부호화 규칙

참고문헌 [5]의 표 1과 표 2는 문자의 전송측면에서 문자를 최적 원천부호화하기 위한 규칙이다. 이 원천부호화 규칙은 OSI 물리계층의 회선부호화 과정에서 수행되는 HDB-3 스크램블링을 고려한 것이다. HDB-3 스크램블링 기능은 원천부호기로부터 출력되는 연속된 4비트이상의 비트열이 회선부호기에 입력될 때, 이 연속 네 개의 0의 비트를 정해진 연속 0의 비트가 아닌 비트열로 대체시킨다. 참고문헌 [5]에서 제시하는 원천부호화 규칙은 문자를 4x4비트 형태로 원천 부호화할 때 사용빈도가 높은 문자에는 비트 0이 연속하여 네 개이상 발생되지 않도록 부호화하는 규칙이다.

3-2 UTF-8부호와 원천부호화 규칙

원천부호화 규칙[5]은 UTF-8 부호에도 적용된다. 원천부호화 규칙에 부합되는 유니코드는 UTF-8 부호로 변환되어도 원천부호화 규칙에 부합됨을 의미한다. UTF-8 부호에서 연속된 네 개 이상의 "0"의 비트가 발생되지 않기 위해서는 원천부호화 규칙에 따라 다음과 같이 부호화되어야 한다. 첫째, 표 1에서 aaaa 네 비트는 16진수 0과 1이 오지 않도록 해야 한다. 이것은 원천부호화 규칙에 의해 16진수 E가 16진수 0과 1과 조합이 제한되기 때문이다. 둘째, bbbb의 네 비트도 16진수 0과 1이 되지 않아야 한다. bbbb가 16진수 4가 될 때는 cccc가 16진수 1, 2, 3과 조합되지 않아야 한다. bbbb가 16진수 8인 경우에는 cccc가 16진수 1, 2, 3, 4, 5, 6, 7이 되지 않아야 한다. 세 번째 비트열 cccc에 4 또는 c가 될 때는 네번째 비트열 dddd가 16진수 1, 2, 3이 되지 않아야 한다. cccc가 16진수 8이 될 때는 dddd가 16진수 1, 2, 3, 4, 5, 6, 7이 되지 않아야 한다. cccc가 16진수 2, 6, A, E일 때에는 dddd가 1이 되지 않아야 한다.

3-3 유니코드의 한글낱자 원천부호화 현황

표 2와 표 3은 유니코드 BMP에 있는 한글낱자(hangul jamo) [11]와 호환용 한글낱자[12]에 대한 부호표이다. 표 2의 한글낱자 유니코드는 1993년 6월에 유니코드 버전 1.1(ISO/IEC 10646-1:1993)로 추가되었다[7]. 표 3의 호환용 한글 낱자는 1991년 10월에 유니코드 버전 1.0에 추가되었다[13]. 표 2에서 U+1100-U+115E까지는 초성, U+1161-11A7까지는 중성, U+11A8-U+11FF까지는 종성에 대한 부호이다. 이 부호표는 한글 옛체를 포함하고 있고 초성, 중성, 종성을 ㄱ, ㄴ, ㅏ, ㅑ 의 자음과 모음순으로 부호를 부여하고 있다. 표 3은 유니코드상의 호환용 한글 글자에 대한 부호표이다. 이 부호표는 자음과 모음에 대한 것으로 현재는 사용하지 않는 옛글을 포함하고 있다. 표 2와 표 3은 두벌식옛글 자판으로 입력할 수 있다. 다만 표 3의 317F의 글자는 자판에서 직접 입력할 수 없다. 한편 표 2와 표 3에는 통상적으로 사용되지 않는 자음과 모음이 상당히 포함되어 있고 또한 자음과 모음이 사전적 순서로 배열되어 있다. 한편 표 2는 자음과 모음이 초성, 중성, 종성으로 부호화되어 있지만 표 3은 자음과 모음만으로 부호화되어 있다. 표 2와 표 3에서 보는 바와 같이 유니코드이 한글낱자부호 배열은 데이터의 전송효율 측면을 고려하지 않은 사전적 단수 배열임을 보여주고 있다.

3-4 한글낱자 부호와 원천부호화 규칙

표 2는 유니코드의 한글낱자에 대한 부호표이다. 이 부호표는 한글의 자음과 모음 낱자를 초성, 중성, 종성으로 하였다. 삽입될 표는 파일에 삽입하고 표를 삽입한 후에 위치에서 일반글자처럼 선택하고 여백캡션에서 모든 여백을 0으로 정한다. 표 안의 내용은 돋움, 7.5포인트, 장평100, 자간-3 정렬은 양쪽정렬로, 줄간격은 120%로 한다. 이 표 2에서 보듯이 이 부호표의 BMP 부호판은 1100~11FF이다. 즉 전체 부호수는 256개가 된다. 이 표에서 빗금 그물표시 부호는 스크램블링이 발생하는 것들이다. 결과적으로 총 256개의 부호에서 63개의 부호가 원천부호화될 때 스크램블링이 발생한다. 그러므로 스크램블링이 발생하지 않는 부호는 총 193개가 된다. 예를 들어 "ㄱ" 의 경우가 부호가 16진수 1100으로 2진수로 표시하면 0001000100000000이 된다. "ㄴ"의 경우에는 부호가 16진수로 1102로서 2진수로는 0001000100000010이 된다. 즉 이러한 원천 부호들이 물리계층에 입력될 때 물리계층의 회선 부호화기에서는 이 부호를 스크램블링해야만 한다.

3-5 호환용한글낱자부호와 원천부호화규칙

표 3은 유니코드 호환용 한글낱자에 대한 부호표이다. 이 부호표의 BMP 부호판은 3130~318F까지이다. 이 호환용 한글낱자 부호표는 초성, 중성과 종성으로 구분하지 않고 자음과 모음으로 구분하여 부호화하였다. 이 표 3의 부호에서 빗금표시 부분은 스크램블링이 발생하는 것들이다. 그러므로 총 부호수는 96개중에서 스크램블링이 발생하는 부호는 총 28개가 된다. 결과적으로 이 표 3에서 스크램블링이 발생하지 않는 부호는 총 68개가 된다. 예를들어 이 표에서 "ㅁ"의 부호는 16진수 3141로서 2진수로 표시하면 0011000101000001이 된다. 즉 부호내에 "0"의 비트가 연속하여 4개 이상 있음을 알 수 있다. 즉 이 부호는 스크램블링을 하여야 한다.

표 2. 유니코드 한글낱자 부호와 원천부호화 규칙
Table 2. Unicode hangul jamo and source coding rule.

U+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1100	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ	ㅅ	ㅆ	ㅇ	ㅈ	ㅉ	ㅊ	ㅋ
1110	ㅌ	ㅍ	ㅎ	ㅏ	ㅐ	ㅑ	ㅒ	ㅓ	ㅔ	ㅕ	ㅖ	ㅗ	ㅘ	ㅙ	ㅚ	ㅛ
1120	ㅜ	ㅝ	ㅞ	ㅟ	ㅠ	ㅡ	ㅢ	ㅣ	ㅤ	ㅥ	ㅦ	ㅧ	ㅨ	ㅩ	ㅪ	ㅫ
1130	ㅬ	ㅭ	ㅮ	ㅯ	ㅰ	ㅱ	ㅲ	ㅳ	ㅴ	ㅵ	ㅶ	ㅷ	ㅸ	ㅹ	ㅺ	ㅻ
1140	ㅼ	ㅽ	ㅾ	ㅿ	ㆀ	ㆁ	ㆂ	ㆃ	ㆄ	ㆅ	ㆆ	ㆇ	ㆈ	ㆉ	ㆊ	ㆋ
1150	ㆌ	ㆍ	ㆎ	㆏	㆐	㆑	㆒	㆓	㆔	㆕						
1160		ㅏ	ㅐ	ㅑ	ㅒ	ㅓ	ㅔ	ㅕ	ㅖ	ㅗ	ㅘ	ㅙ	ㅚ	ㅛ	ㅜ	ㅝ
1170	ㅞ	ㅟ	ㅠ	ㅡ	ㅢ	ㅣ	ㆍ	ㆎ	ㆍㅡ	ㆍㅣ	ㆎㅡ	ㆎㅣ	ㆍㆍ	ㆍㆍㅡ	ㆍㆍㅣ	ㆍㆎ
1180	ㆎㅣ	ㅑㅣ	ㅕㅣ	ㅗㅣ	ㅛㅣ	ㅜㅣ	ㅠㅣ	ㅡㅣ	ㅣㅣ	ㅏㅡ	ㅐㅡ	ㅓㅡ	ㅔㅡ	ㅗㅡ	ㅘㅡ	ㅙㅡ
1190	ㅚㅡ	ㅝㅡ	ㅞㅡ	ㅟㅡ	ㅠㅡ	ㅡㅡ	ㅢㅡ	ㅣㅡ	ㅏㅣ	ㅐㅣ	ㅑㅣ	ㅒㅣ	ㅓㅣ	ㅔㅣ	ㅕㅣ	ㅖㅣ
11A0	ㅗㅣ	ㅘㅣ	ㅙㅣ	ㅚㅣ	ㅛㅣ	ㅜㅣ	ㅝㅣ	ㅞㅣ	ㅟㅣ	ㅠㅣ	ㅡㅣ	ㅢㅣ	ㅣㅣ	ㆍㅣ	ㆍㆍㅣ	ㆍㆎㅣ
11B0	ㆎㅡ	ㆍㅗ	ㆍㅜ	ㆍㅡ	ㆍㅣ	ㆎㅣ	ㆎㅡ	ㆎㅜ	ㆎㅗ	ㆎㅏ	ㆎㅐ	ㆎㅔ	ㆎㅣ	ㆎㆍ	ㆎㆍㆍ	ㆎㆎ
11C0	ㆎㆍㅣ	ㆎㆎㅣ	ㆎㆎㅡ	ㆎㆎㅜ	ㆎㆎㅗ	ㆎㆎㅏ	ㆎㆎㅐ	ㆎㆎㅔ	ㆎㆎㅣ	ㆎㆎㆍ	ㆎㆎㆍㆍ	ㆎㆎㆍㆎ	ㆎㆎㆎㅣ	ㆎㆎㆎㆍ	ㆎㆎㆎㆍㆍ	ㆎㆎㆎㆎ
11D0	ㆎㆎㆎㆎㅣ	ㆎㆎㆎㆎㆍ	ㆎㆎㆎㆎㆍㆍ	ㆎㆎㆎㆎㆎㅣ	ㆎㆎㆎㆎㆎㆍ	ㆎㆎㆎㆎㆎㆎ	ㆎㆎㆎㆎㆎㆎㅣ	ㆎㆎㆎㆎㆎㆎㆍ	ㆎㆎㆎㆎㆎㆎㆍㆍ	ㆎㆎㆎㆎㆎㆎㆎㅣ	ㆎㆎㆎㆎㆎㆎㆎㆍ	ㆎㆎㆎㆎㆎㆎㆎㆎ	ㆎㆎㆎㆎㆎㆎㆎㆎㅣ	ㆎㆎㆎㆎㆎㆎㆎㆎㆍ	ㆎㆎㆎㆎㆎㆎㆎㆎㆍㆍ	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎ
11E0	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㅣ	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆍ	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆍㆍ	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㅣ	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆍ	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆍㆍ	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㅣ	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆍ	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆍㆍ	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㅣ	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆍ	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆍㆍ	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㅣ	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆍ	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆍㆍ	ㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎㆎ
11F0	ㆎ															

표 3. 유니코드 호환용 한글낱자 부호와 원천부호화 규칙
Table 3. Hangul compatibility jamo and source coding rule.

U+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3130		ㄴ	ㄲ	ㄺ	ㄱ	ㄳ	ㅎㄱ	ㄷ	ㄸ	ㄹ	ㄻ	ㄼ	ㄽ	ㄾ	ㄿ	ㅀ
3140	ㅎㄹ	ㅁ	ㅂ	ㅃ	ㅄ	ㅅ	ㅆ	ㅇ	ㅈ	ㅉ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ	ㅏ
3150	ㅐ	ㅑ	ㅒ	ㅓ	ㅔ	ㅕ	ㅖ	ㅗ	ㅘ	ㅙ	ㅚ	ㅛ	ㅜ	ㅝ	ㅞ	ㅟ
3160	ㅠ	ㅡ	ㅢ	ㅣ		ㄲ	ㄳ	ㄴ	ㄵ	ㄶ	ㄷ	ㄹ	ㄺ	ㄻ	ㄼ	ㄽ
3170	ㄾ	ㄿ	ㅀ	ㅁ	ㅂ	ㅄ	ㅅ	ㅆ	ㅇ	ㅈ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ	ㅏㅣ
3180	ㅐㅣ	ㅑㅣ	ㅒㅣ	ㅓㅣ	ㅔㅣ	ㅕㅣ	ㅖㅣ	ㅗㅣ	ㅘㅣ	ㅙㅣ	ㅚㅣ	ㅛㅣ	ㅜㅣ	ㅝㅣ	ㅞㅣ	

표 4. 표 2의 한글낱자 사용 빈도율 및 빈도수 [4]
Table 4. Using frequency and rate of Hangul Jamo in table 2[4].

First Consonant			Vowel			Final Consonant		
Consonant	Using Frequency Rate%	Using Frequency	Vowel	Using Frequency Rate%	Using Frequency	Consonant	Using Frequency Rate%	Using Frequency
ㄱ	5.372	1,171,038	ㅏ	8.835	1,926,007	ㄱ	1.888	411,538
ㄲ	0.298	64,997	ㅐ	1.880	409,768	ㄲ	0.028	6,068
ㄴ	2.738	596,893	ㅑ	0.292	63,728	ㄳ	0.002	357
ㄷ	3.673	800,785	ㅒ	0.017	3,769	ㄴ	6.092	1,328,090
ㄸ	0.333	72,604	ㅓ	4.325	942,819	ㄵ	0.013	2,790
ㄹ	2.799	610,211	ㅔ	1.806	393,695	ㄶ	0.139	30,376
ㅁ	1.971	429,661	ㅕ	1.938	422,490	ㄷ	0.085	18,535
ㅂ	1.647	359,029	ㅖ	0.204	443,94	ㄹ	3.564	776,891
ㅃ	0.074	16,037	ㅗ	3.955	862,073	ㄺ	0.034	7,330
ㅅ	3.470	756,477	ㅘ	0.746	162,656	ㄻ	0.020	4,344
ㅆ	0.134	29,184	ㅙ	0.051	11,215	ㄼ	0.010	2,183
ㅇ	9.659	2,105,587	ㅚ	0.458	99,794	ㄽ	0.000	5
ㅈ	3.434	748,509	ㅛ	0.434	94,576	ㄾ	0.001	131
ㅉ	0.094	20,434	ㅜ	2.718	592,419	ㄿ	0.000	78
ㅊ	0.919	200266	ㅝ	0.263	57,373	ㅀ	0.014	3,023
ㅋ	0.221	48097	ㅞ	0.009	2,035	ㅁ	1.186	258,542
ㅌ	0.489	106679	ㅟ	0.218	47,506	ㅂ	0.613	133,729
ㅍ	0.454	98986	ㅠ	0.216	47,135	ㅄ	0.119	25,868
ㅎ	2.914	635299	ㅡ	5.285	1,151,976	ㅅ	0.526	114,592
			ㅢ	0.825	179,818	ㅆ	1.005	219,186
			ㅣ	6.218	1,355,527	ㅇ	2.831	617,205
						ㅈ	0.079	17,173
						ㅊ	0.046	9,944
						ㅋ	0.001	302
						ㅌ	0.116	25,341
						ㅍ	0.078	16,938
						ㅎ	0.124	26,948

3-6 한글낱자 사용 빈도율 및 빈도

표 4는 한글낱자에 대한 사용빈도율과 빈도수를 보여 주는 것이다. 초성 19자, 중성 21자, 종성 27자로 총 67자이다[4]. 사용빈도가 가장 높은 것과 가장 낮은 순서로 볼 때 초성은 "ㅇ", "ㅃ", 중성은 "ㅏ", "ㅖ" 종성은 "ㄴ", "ㄺ"이다.

Ⅳ. Unicode와 UTF-8 부호체계 분석

표 5는 참고문헌[5] 표 1에 의하여 Unicode 한글낱자와 이에 대한 UTF-8부호체계에서 발생하는 스크램블링 발생 현황을 분석한 것이다. 이 표에서 A는 자모의 사용 빈도율로서 표 4에 의한 것이다. B는 참고문헌 [5] 표 1에 의하여 분석된 각 자모에서 부호에서 발생하는 스크램블링이 횟수이다. 초성 "ㄱ"의 경우 유니코드에서는 스크램블링이 2번 발생하고 UTF-8에서는 3번 발생한다. C는 각 자모의 스크램블링 발생 횟수에 의하여 스크

램블링 발생 빈도율을 사용 빈도율로 나타낸 것이다. 예를 들어 초성“ㄱ”의 경우에 사용 빈도율이 5.37%이고 스크램블링 발생 횟수가 2번 이므로 총 스크램블링 발생율은 10.74%가 된다. 이 표 5에서와 같이 한글낱자 67개의 자모중에서 스크램블링이 발생하는 횟수는 유니코드에서 24개, UTF-8에서 148개 발생하였다. 즉, 유니코드를 UTF-8로 변환하면서 스크램블링이 124개가 더 발생하였다. 결과적으로 유니코드를 UTF-8로 변환하면서 스크램블링이 517% 증가한 것이다. 스크램블링 발생값을 각 자모의 사용 빈도율에 의하여 계산할 경우에 유니코드에서의 스크램블링 발생율은 51.9%, UTF-8은 228.4%발생하였다. 즉, 유니코드를 UTF-8로 변환하면서 사용 빈도율에 의한 스크램블링이 176.5% 더 발생하였다. 결과적으로 유니코드를 UTF-8로 변환하면서 스크램블링이 340% 증가한 것이다.

표 5. Unicode 한글 낱자와 UTF-8 부호의 스크램블링 발생 원천천부호화 규칙[5] 표 1 적합성 비교 (A: 사용빈도율, B: 스크램블링 발생횟수 C: 스크램블링 발생율

Table 5. Aptness of Hangul Jamo between Unicode and its UTF-8 Code for source coding rule[5] Table 1. (A: jamo using frequency rate B: scrambling frequency C: scrambling frequency rate)

Hangul jamo		Unicode		UTF-8	
Jamo	A%	Bea	C%	Bea	C%
ㄱ	5.37	2	10.74	3	16.12
ㄲ	0.30	1	0.30	3	0.89
ㄴ	2.74	1	2.74	3	8.21
ㄷ	3.67	1	3.67	3	11.02
ㄸ	0.33	1	0.33	3	1.00
ㄹ	2.80	1	2.80	3	8.40
ㅁ	1.97	1	1.97	3	5.91
ㅂ	1.65	1	1.65	3	4.94
ㅃ	0.07	1	0.07	3	0.22
ㅅ	3.47	1	3.47	2	6.94
ㅆ	0.13	1	0.13	2	0.27
ㅇ	9.66	1	9.66	2	19.32
ㅈ	3.43	1	3.43	2	6.87
ㅉ	0.09	1	0.09	2	0.19
ㅊ	0.92	1	0.92	2	1.84
ㅋ	0.22	1	0.22	2	0.44
ㅌ	0.49	1	0.49	2	0.98
ㅍ	0.45	0	0.00	3	1.36
ㅎ	2.91	0	0.00	2	5.83
ㅏ	8.84	1	8.84	3	26.51
ㅐ	1.88	0	0.00	2	3.76
ㅑ	0.29	0	0.00	2	0.58
ㅒ	0.02	0	0.00	2	0.03
ㅓ	4.33	0	0.00	2	8.65
ㅔ	1.81	0	0.00	2	3.61
ㅕ	1.94	0	0.00	2	3.88
ㅖ	0.20	0	0.00	2	0.41
ㅗ	3.95	0	0.00	2	7.91
ㅘ	0.75	0	0.00	2	1.49
ㅙ	0.05	0	0.00	2	0.10
ㅚ	0.46	0	0.00	2	0.92
ㅛ	0.43	0	0.00	2	0.87
ㅜ	2.72	0	0.00	2	5.44
ㅝ	0.26	0	0.00	2	0.53
ㅞ	0.01	1	0.01	2	0.02
ㅟ	0.22	0	0.00	2	0.44
ㅠ	0.22	0	0.00	2	0.43
ㅡ	5.28	0	0.00	2	10.57
ㅢ	0.82	0	0.00	2	1.65
ㅣ	6.22	0	0.00	2	12.44
ㄱ	1.89	0	0.00	2	3.78
ㄲ	0.03	0	0.00	2	0.06
ㄳ	0.00	0	0.00	2	0.00
ㄴ	6.09	0	0.00	2	12.18
ㄵ	0.01	0	0.00	2	0.03
ㄶ	0.14	0	0.00	2	0.28
ㄷ	0.09	0	0.00	2	0.17
ㄹ	3.56	0	0.00	2	7.13
ㄺ	0.03	1	0.03	2	0.07
ㄻ	0.02	0	0.00	3	0.06
ㄼ	0.01	0	0.00	2	0.02
ㄽ	0.00	0	0.00	2	0.00
ㄾ	0.00	0	0.00	2	0.00
ㄿ	0.00	0	0.00	2	0.00
ㅀ	0.01	0	0.00	2	0.03
ㅁ	1.19	0	0.00	2	2.37
ㅂ	0.61	0	0.00	2	1.23
ㅄ	0.12	0	0.00	2	0.24
ㅅ	0.53	0	0.00	2	1.05
ㅆ	1.01	0	0.00	2	2.01
ㅇ	2.83	0	0.00	2	5.66
ㅈ	0.08	0	0.00	2	0.16
ㅊ	0.05	0	0.00	2	0.09
ㅋ	0.00	0	0.00	2	0.00
ㅌ	0.12	1	0.12	2	0.23
ㅍ	0.08	1	0.08	3	0.23
ㅎ	0.12	1	0.12	3	0.37
Total	100	24	51.9	148	228.4

표 6은 참고문헌[5] 표 1에 의하여 Unicode 호환용 한글낱자와 이에 대한 UTF-8부호체계에서 발생하는 스크램블링발생 현황을 분석한 것이다. 이 표에서 A는 자모의 사용 빈도율로서 표 4에 의한 것이다. B는 표 1의 각 자모에서 스크램블링이 발생하는 횟수이다. 초성 “ㄱ”의 경우 유니코드에서는 스크램블링이 발생하지 않고 UTF-8에서는 1번 발생한다. C는 각 자모의 스크램블링 발생 횟수에 의하여 스크램블링발생 빈도율을 사용 빈도율로 나타낸 것이다. 예를 들어 초성“ㄱ”의 경우에 사용 빈도율이 5.37%이지만 유니코드에서는 스크램블링이 발생하지 않으므로 스크램블링 발생율은 0%가 된다.

표 6. Unicode 호환용 한글 낱자와 UTF-8 부호의 원천천부호화 규칙 적합성 비교([5] 표 1)

Table 6. Aptness of unicode hangul compatibility jamo and its UTF-8 code for source coding rule[5] Table 1. (A: jamo using frequency rate B: scrambling frequency C: scrambling frequency rate)

Hangul jamo		.Unicode		UTF-8	
Jamo	A%	Bea	C%	Bea	C%
ㄱ	5.37	0	0.00	1	5.37
ㄲ	0.30	0	0.00	1	0.30
ㄴ	2.74	0	0.00	1	2.74
ㄷ	3.67	0	0.00	1	3.67
ㄸ	0.33	0	0.00	1	0.33
ㄹ	2.80	0	0.00	1	2.80
ㅁ	1.97	1	1.97	1	1.97
ㅂ	1.65	1	1.65	2	3.29
ㅃ	0.07	1	0.07	2	0.15
ㅅ	3.47	0	0.00	2	6.94
ㅆ	0.13	0	0.00	2	0.27
ㅇ	9.66	0	0.00	2	19.32
ㅈ	3.43	0	0.00	2	6.87
ㅉ	0.09	0	0.00	1	0.09
ㅊ	0.92	0	0.00	1	0.92
ㅋ	0.22	0	0.00	1	0.22
ㅌ	0.49	0	0.00	1	0.49
ㅍ	0.45	0	0.00	1	0.45
ㅎ	2.91	0	0.00	1	2.91
ㅏ	8.84	0	0.00	1	8.84
ㅐ	1.88	1	1.88	1	1.88
ㅑ	0.29	0	0.00	2	0.58
ㅒ	0.02	0	0.00	1	0.02
ㅓ	4.33	0	0.00	1	4.33
ㅔ	1.81	0	0.00	1	1.81
ㅕ	1.94	0	0.00	1	1.94
ㅖ	0.20	0	0.00	1	0.20
ㅗ	3.95	0	0.00	1	3.95
ㅘ	0.75	0	0.00	1	0.75
ㅙ	0.05	0	0.00	1	0.05
ㅚ	0.46	0	0.00	1	0.46
ㅛ	0.43	0	0.00	1	0.43
ㅜ	2.72	0	0.00	1	2.72
ㅝ	0.26	0	0.00	1	0.26
ㅞ	0.01	0	0.00	1	0.01
ㅟ	0.22	0	0.00	1	0.22
ㅠ	0.22	1	0.22	1	0.22
ㅡ	5.28	1	5.28	2	10.57
ㅢ	0.82	1	0.82	2	1.65
ㅣ	6.22	0	0.00	1	6.22
ㄱ	1.89	0	0.00	1	1.89
ㄲ	0.03	0	0.00	1	0.03
ㄳ	0.00	0	0.00	1	0.00
ㄴ	6.09	0	0.00	1	6.09
ㄵ	0.01	0	0.00	1	0.01
ㄶ	0.14	0	0.00	1	0.14
ㄷ	0.09	0	0.00	1	0.09
ㄹ	3.56	0	0.00	1	3.56
ㄺ	0.03	0	0.00	1	0.03
ㄻ	0.02	0	0.00	1	0.02
ㄼ	0.01	0	0.00	1	0.01
ㄽ	0.00	0	0.00	1	0.00
ㄾ	0.00	0	0.00	1	0.00
ㄿ	0.00	0	0.00	1	0.00
ㅀ	0.01	0	0.00	1	0.01
ㅁ	1.19	1	1.19	2	2.37
ㅂ	0.61	1	0.61	2	1.23
ㅄ	0.12	0	0.00	2	0.24
ㅅ	0.53	0	0.00	2	1.05
ㅆ	1.01	0	0.00	2	2.01
ㅇ	2.83	0	0.00	2	5.66
ㅈ	0.08	0	0.00	2	0.16
ㅊ	0.05	0	0.00	1	0.05
ㅋ	0.00	0	0.00	1	0.00
ㅌ	0.12	0	0.00	1	0.12
ㅍ	0.08	0	0.00	1	0.08
ㅎ	0.12	1	0.12	1	0.12
Total	100	10	13.82	83	131.2

UTF-8에서는 스크램블링이 1회 발생하므로 스크램블링발생율은 5.37%가 된다. 이 표 6에서와 같이 호환용 한글낱자 67개의 자모중에서 스크램블링이 발생하는 횟수는 유니코드에서 10개, UTF-8에서 83개 발생하였다. 즉, 유니코드를 UTF-8로 변환하면서 스크램블링이 73개가 더 발생하였다. 결과적으로 유니코드를 UTF-8로 변환하면서 스크램블링이 730% 증가한 것이다. 스크램블링 발생값을 각 자모의 사용 빈도율에 의하여 계산할 경우에 유니코드에서의 스크램블링 발생율은 13.82%, UTF-8은 131.34%발생하였다. 즉, 유니코드를 UTF-8로 변환하면서 사용 빈도율에 의한 스크램블링이 117.5% 더 발생하였다. 결과적으로 유니코드를 UTF-8로 변환하면서 스크램블링이 851% 증가한 것이다.

Ⅴ. 결 론

본 논문은 유니코드 한글낱자 부호와 유니코드 호환용 한글낱자의 부호가 UTF-8부호로 변환될 때 회선부호화과정에서 일어나는 스크램블링 발생에 어떠한 영향을 미치는 지를 연구하였다. 한글낱자의 경우에 한글낱자 67개의 자모중에서 스크램블링이 발생하는 횟수는 유니코드에서 24개, UTF-8에서 148개 발생하였다. 결과적으로 유니코드를 UTF-8로 변환하면서 스크램블링 발생 횟수가 517% 증가한 것이다. 스크램블링 발생값을 각 자모의 사용 빈도율에 의하여 계산할 경우에 유니코드에서의 스크램블링 발생율은 51.9%, UTF-8은 228.4%발생하였

다. 결과적으로 유니코드를 UTF-8로 변환하면서 스크램블링 발생빈도율은 340% 증가하였다. 호환용 한글낱자의 경우에는 호환용 한글낱자 67개의 자모중에서 스크램블링이 발생하는 횟수는 유니코드에서 10개, UTF-8에서 83개 발생하였다. 결과적으로 유니코드를 UTF-8로 변환하면서 스크램블링 발생횟수는 730% 증가하였다.한 것이다. 스크램블링 발생값을 각 자모의 사용 빈도율에 의하여 계산할 경우에 유 니코드에서의 스크램블링 발생 빈도 율은 13.82%, UTF-8은 131.34%발생하였다. 결과적으로 유니코드를 UTF-8로 변환하면서 스크램블링 발생 사용빈도율은 851% 증가한 것이다. 이러한 결과가 한글낱자와 호환용한글낱자에서 발생한 것은 유니코드가 UTF-8부호로 변환될 때 가장 많은 스크램블링이 발생하는 유니코드체계 11XX와 31XX로 한글낱자와 호환용 한글낱자부호체계가 되어 있기 때문이었다. 따라서 현재의 유니코드의 한글낱자와 호환용 한글낱자의 부호 배열판을 다른 부호 배열판으로 바꾸지 않는 한 이러한 문제를 해결하는데 큰 어려움이 있을 것으로 분석되었다. 본 연구결과를 토대로 향후 스크램블링이 최소화되는 한글낱자와 호환용 한글낱자의 부호체계의 연구가 필요하다. 아울러 유니코드상의 한글글자마디부호체계의 스크램블링 빈도에 대한 연구도 필요하다.

참고 문헌

[1] B. A. Forouzan, *Data communications,* New York, NY: McGraw-Hill, 2008.

[2] P. V. Sreekanth, *Digital Microwave Communication Systems,* Hyderguda, India: Universities press(india) Private Limited, 2003.

[3] P. C. Gupta, *Data Communications and Computer Networks,* 2nd ed, Delhi, India: PHI Learning Private Limited, 2014.

[4] H. S. Kim, *Korean Use Frequency Survey,* Seoul, Korea: The National Institute of the Korean Language, 2005.

[5] W. Hong, “Coding rule of characters by 2 bytes with 4x4 bits to improve the transmission efficiency in data communications,” *The Journal of Korea Navigation Institute*, Vol. 15, No. 5, pp. 745-751, Oct. 2011.

[6] J. Alipranda, *The Unicode Standard,* Boston, MA: Addison Wesley, 2004.

[7] The Unicode Consortium. Components of The Unicode St andard Version 1.0.0 [Internet]. Available: http://www.unicode.org/versions/components-1.0.0.html

[8] F. Yergeau, *UTF-8, a Transformation Format of Unicode and ISO 10646,* Montreal, Canada: Alis Technologies, 1996.

[9] The Unicode Consortium. UTF-8 encoding table and Unicode characters page with code points U+1100 to U+11FF. [Internet]. Available: http://www.utf8-chartable.de/unicode-utf8-table.pl?start=4352

[10] The Unicode Consortium. UTF-8 encoding table and Unicode characters page with code points U+3100 to U+31FF. [Internet]. Available: http://www.utf8-chartable.de/unicode-utf8-table.pl?start=12544

[11] The Unicode Consortium. Components of The Unicode Standard Version 1.0.0 [Internet]. Available: http://www.unicode.org/charts/PDF/U1000.pdf

[12] The Unicode Consortium. Components of The Unicode Standard Version 1.0.0 [Internet]. Available: http://www.unicode.org/charts/PDF/U3000.pdf

[13] The Unicode Consortium. Unicode 7.0.0 Released: 2014 June 16 [Internet]. Available: http://www.unicode.org/versions/Unicode7.0.0/

홍 완 표 (Wan-pyo Hong)

1991년 : 서울과학기술대학교 전자공학과 (공학사),
1994년 : 연세대학교 공학대학원 전자공학전공 (공학석사)
1999년 : 광운대학교 대학원 전자공학과 (공학박사),
1990년 : 전기통신기술사합격
1991년 : 정보통신부 5급특별채용고시합격 본부 통신정책실, 전파방송관리국, 정보화기획실
1997년 : 삼성전자(주) 통신사업부 전송영업그룹장,
1999년 : 광운대학교 연구전담교수
2000년 : 한국정보통신기술사협회장,
2002년 : 한세대학교 정보통신공학과 교수
2014년 : USC 동북아언어문화학과 방문학자
※ 관심분야 : 위성통신방송, 문자코딩, 통신정책.