

HDB-3 스크램블링과 HDLC 프로토콜에서 효율적 문자부호 전송을 위한 문자부호 작성 규칙

Composition Rule of Character Codes to efficiently transmit in HDLC Protocol with HDB-3 Scrambling

홍완표*

Wan-Pyo Hong*

요 약

정보 기기 내에서 문자를 어떠한 규칙에 의해 부호화하는가에 따라 데이터 전송효율을 제고시킬 수 있다. 본 논문은 데이터 전송 효율면에서 OSI 표현계층에서 이루어지는 문자부호화 원칙을 제시하였다. 본 논문에서 제시하는 문자부호화 원칙은 두가지점을 고려하여 제시되었다 첫째는 OSI 데이터링크 계층의 HDLC와 PPP 프로토콜의 플래그 필드를 구성하는 비트열이다. 둘째는 OSI 물리계층에서 이루어지는 HDB-3 스크램블링 방식이다. 첫 번째 고려사항에서는 원천부호에 비트“1”이 연속하여 다섯 개 이상 발생치 않도록 하는 것이다. 두 번째 고려사항에서는 원천부호에 비트“0”이 연속하여 네 개 이상 발생치 않도록 하는 것이다. 본 논문에서 제시하고 있는 원천부호화 규칙에 의하여 문자를 원천부호화 할 때에 데이터의 전송효율을 제고시키게 된다.

Abstract

In the transmission system, the transmission efficiency can be increased by applying the proper character coding method. This paper is to show the character coding rule in computer and information equipment etc to improve the transmission efficiency in datacommunications. The character coding rule showing in this paper was suggested by considering the two view points. First one is FLAG bits in HDLC and PPP protocol on datalink layer in OSI 7 layer. Second one is one of the scrambling method, HDB-3, on physical layer in OSI 7 layer. The former is to prevent the sequence of over 5bits of "1" in the character codes. The latter is to prevent the sequence of over 4bits of "0" in the character codes.

Key words : Source coding, Scrambling, HDB-3, HDLC, Character code

I. 서 론

데이터통신에 있어서 OSI의 응용계층에서 입력된 문자는 표현계층에서 일정한 부호(code)로 원천 부호

화(source coding)된다. ASCII[1][2][3], EBCDIC[4], Unicode 등은 대표적인 원천부호화 규격이다. 이러한 문자코드는 조합형 또는 완성형의 형태로 처리된다. 영어는 유니코드 조합형[5], 한글은 유니코드 조합형과 완성형코드[6]로 되어 있다. 이 원천 부호화된 부

* 한세대학교 정보통신공학과(Dept. of Information & Telecommunication Hansei University)

· 제1저자 (First Author) : 홍완표
· 투고일자 : 2012년 8월 13일
· 심사(수정)일자 : 2012년 8월 17일 (수정일자 : 2012년 10월 17일)
· 게재일자 : 2012년 10월 30일

호가 네트워크를 통하여 전송되기 위해서는 네트워크 계층에서 패킷화 된다. 이 패킷화된 부호는 데이터 링크 계층에서 프레임화 된다. HDLC와 PPP는 데이터 링크 계층에서 프레임화되는 대표적인 프로토콜이다. HDLC와 PPP 프로토콜의 프레임의 초기와 종기 필드는 플래그(FLAG)필드로 구성된다. 이 플래그 필드는 프레임의 시작과 끝을 인식하기 위한 것으로 고유한 비트열로 구성된다. 그러므로 프레임내에 이 플래그 비트열과 같은 비트열을 갖는 부호가 있을 경우에는 다른 비트열로 부호를 바꾸어주어야 한다. 이와 같이 데이터링크 계층에서 프레임화된 부호는 물리계층에서 전송로에 적합한 신호로 부호화된다. 이것을 회선부호화(line coding)라고 한다.

프레임내에 있는 부호를 회선 부호화하는 목적은 첫째 연속되는 동일비트로 송신 시 수신측에서 비트간의 동기를 맞출 수 없어 데이터를 검파할 수 없게 되는 것을 방지하는 것이다. 둘째, 원천신호의 직류성분을 원거리 전송에 적합한 교류적 성분으로 변경하여 주는 것이다. 회선 부호화하는 방식 중에 AMI(alternate Mark Space)방식이 있다. 이 방식은 비트열중에서 "0"이 연속될 경우에 이에 대한 대책이 없다. 즉 회선 부호화의 본래 목적을 충분히 달성할 수 없다. 그러므로 이에 대한 대책으로 연속"0"의 비트열을 인위적으로 비연속"0"의 비트열로 바꾸어 주는 방법을 사용하고 있다. 이러한 방식을 스크램블링(scrambling)이라고 한다. 대표적인 스크램블링방식으로 B8ZS[7][8]방식과 HDB-3[9][10]방식이 있다. 전자는 연속 8개의 "0"을 방지하는 것이고 후자는 연속 4개의 "0"의 연속을 방지하는 것이다. 본 논문에서는 이상에서 언급한 데이터링크 계층과 물리계층에서 발생하는 플래그 비트열의 변경과 스크램블링이 발생하는 경우를 감소시킴으로서 데이터의 전송효율을 제고시키고자 하는 것이다. 그러한 방법으로 표현계층의 원천부호화과정에서 이러한 목적을 달성할 수 있도록 하는 원천부호화에 대한 규칙을 제시하였다.

II. 원천부호화 표준 규격

2-1 원천 부호화 표준 현황

한국기술표준원은 2004년도에 정보교환용 부호계(한글 및 한자)에 대한 표준규격 KSX 1001을 개정, 공표하였다[11]. 이 규격은 정보 처리 및 데이터를 전송하는 시스템에서 정보 교환에 사용하는 부호계의 표현 형식에 대한 것을 규정하고 있다. 이 규격에서는 문자, 도형문자, 특수문자, 숫자, 한글날자, 한글글자마디, 로마문자, 한자, 그리스문자 그리고 바이트 등에 대한 정의를 하고 있다. 한글 날자는 한글 글자 마디를 이루는 닿소리, 홀소리 글자 또는 첫소리, 가운데소리, 끝소리 글자("?" 이나 "나" 같은 겹글자도 하나의 날자로 여긴다.)를 가리킨다. 한글 날자는 닿소리 글자 30자, 홀소리 글자 21자, "채움"(끝소리 글자 없음 표시) 1문자와 고어 42문자, 총 94문자로 한다[11][12]. 한글 글자 마디는 첫소리와 가운데소리 글자, 또는 첫소리, 가운데소리, 끝소리 글자로 이루어진 한글의 단위이다. 한글 글자 마디는 사용 빈도에 의하여 선정된 한글 글자 마디 2,350자로 하고 있다. 한 문자의 코드는 행과 열로 구성된 2바이트조합으로 구성된다. 바이트는 정보 교환의 편의상 한 단위로 취급되는 8개의 비트를 의미한다.[11][12]

2-2 문자 부호계 규격

이 규격에서는 부호계의 단위를 2바이트로 하고 있다. 각각의 바이트는 정보 교환용 부호계의 확장법(ISO/IEC2022)에 따라 7비트 또는 8비트로 하고 있다. 2바이트 중 앞의 바이트를 첫째 바이트라고 하고 뒤의 바이트를 둘째 바이트라 하고 있다. 도형문자에 대한 부호계열은 7/8비트열 바이트 모두 b7-b1비트열로 구성된다. 따라서 8비트열 바이트의 경우에는 최상위 비트인 b8비트가 항상 "1"이 되게 한다. 이 규격에 포함되지 않는 한글 글자 마디는 첫소리, 가운데소리, 끝소리 글자로 분류하고 각 부분에 해당되는 특수 문자 영역의 한글 날자 부호 값을 정보 교환에 사용한다. 한글 날자만의 정보 교환과 구분하기 위하여 한글 글자 마디마다 맨 앞에 "채움" 문자를 1개씩 추가하도록 하고 있다[11][12].

2-3 문자부호의 분류와 배열

참고문헌 [11]부속서2에서는 도형문자의 분류와

배열에 대하여 규정하고 있다. 사용 빈도가 높은 특수 문자, 숫자, 한글 낱자, 외국 문자 및 괄선 조각은 제1행~제12행에 배열하고 있다. 한글글자마디는 총 2350자를 가나다순으로 제16행~제40행에 배열하고 있다.

2-4 2바이트 조합형 부호계 규칙

참고문헌 [11]부속서3에는 기본 부호계인 2바이트 완성형 부호계의 보조 부호계로서 2바이트 조합형 부호계를 규정하고 있다. 한글은 [11]부속서 3 표 2에 규정된 첫소리 글자 19자, 가운뎃소리 글자 21자, 끝소리 글자 27자로 조합 가능한, 모든 현대 한글 글자마디(11 172자) 및 현대 한글 낱자(67자)에 대한 것이다. 한글글자마디에 대한 코드배치영역은 첫번째 바이트는 84~D3, 두 번째 바이트는 41~7E, 81~FE로 규정하고 있다. 한글글자마디에 대한 부호값은 2바이트 내에 첫소리 글자 5비트, 가운뎃소리 글자 5비트, 끝소리 글자 5비트로 하여 한글 낱자를 조합하여 표현한 값으로 하고 있다[12]. 각 한글 낱자의 순서는 최상위 비트(MSB)를 1로 하고 첫소리, 가운뎃소리, 끝소리 글자가 순서대로 나오도록 구성하고 있다.

2-5 7비트 한글낱자 부호계 규정

참고문헌 [11]규격 부속서 4에는 한글낱자 7비트 부호계에 대하여 규정하고 있다. 이 규격은 권장규격으로 원칙적으로 정보교환용으로는 사용하지 않는 것으로 규정하고 있다. 7비트 한글 낱자 부호값은 한글 낱자 1자를 표현한다. 즉, 한개의 글자마디는 첫소리 바이트, 가운뎃소리 바이트, 끝소리 바이트의 세개의 바이트로 조합되어 나타난다. 7비트 한글낱자 부호표에서 “채움”의 용법은 첫소리, 가운뎃소리, 끝소리 글자가 없을 때 각각 그 자리에 넣는 부호이다[12].

이상에서와 같이 [11]의 정보 교환용 부호계는 국제 표준화 기구 (ISO)의 정보 교환용 부호계의 확장법(ISO/IEC 2022)을 따르는 2바이트 부호계로 규정하고 있다. 부호 값의 범위는 2121-7E7EH까지 8836문자를 수용할 수 있도록 구성하고 있다. 2바이트 조합형 부호계에서 모아 쓴 한글의 부호값은 KS X

1003(KSC 5636)에 규정되어 있는 7비트 부호계와 8비트 부호계의 기능 문자와 중복되지 않도록 하고 있다. 첫소리와 가운뎃소리 글자 자리에도 “채움” 문자 부호를 배열하여 2바이트 조합형 부호계에서 한글 낱자 부호값도 한글 글자 마디와 같은 방법으로 만들 수 있도록 하고 있다[11][12]. 또한 정보 교환용 한글 부호계에 포함되지 않은 한글 글자 마디는 그 출현 빈도가 매우 낮을 것으로 보고 보통 한글 낱자와 구분하기 위하여 각 한글 글자 마디마다 맨 앞에 “채움” 문자 1개씩 덧붙여 교환하도록 하고 있다. 도형문자의 한글글자마디 부호표는 총2350자를 가나다순으로 배열하고 있다. 이와 같이 현재의 한글부호계 규격은 국제표준 부호배열 규격에 맞추어 가나다 또는 ㄱ, ㄴ, ㄷ 등의 순서로 단순 배열하고 있음을 알 수 있다[11][12]

III. 채우기(Stuffing)와 뒤섞기(Scrambling)

3-1 채우기(Stuffing)

채우기는 데이터링크 계층의 HDLC[13][14][15]와 PPP[13][16]프로토콜의 프레임에 적용되는 기술이다.

가. HDLC

HDLC 프로토콜은 비트중심 동기전송방식에 적용된다. 이 프로토콜을 구성하는 프레임의 첫 번째와 마지막 필드를 플래그(flag)필드라고 한다. 이 플래그 필드는 “01111110”(7E hex)로 구성된다. HDLC 프로토콜은 정보 프레임(I-frame), 감시프레임(S-frame), 무번호 프레임(U-frame) 등 3개의 프레임종을 가지고 있다. 이 세 개의 프레임 모두 프레임의 맨 앞과 뒤에 플래그 필드를 가지고 있다. HDLC 프로토콜에서는 데이터 비트열에 플래그 비트열과 동일한 데이터 비트열이 있을 경우에 두 비트열 간의 오인을 막기 위해 데이터 비트열을 인위적으로 플래그와 다른 형태의 비트열로 바꾼다. 이러한 방법은 데이터 비트열에서 “0”비트 이후에 “1”의 비트가 연속하여 5개 이상 있을 경우 5번째 “1”비트 다음에

“0”비트를 강제로 삽입한다. 즉 7E hex가 7D hex로 바뀌어진다. 그래서 이러한 방법을 비트 채우기(bit stuffing)방식이라고 한다.

나. PPP

PPP프로토콜은 문자중심 비동기 전송방식에 적용된다. 그림1과 같이 이 프레임의 구성하는 첫 번째와 마지막 필드를 플래그 필드라고 한다. 이 플래그 필드는 “01111110” 비트열(7E hex)로 구성된다. 처음 플래그와 마지막 플래그 사이에 플래그 필드와 동일한 비트열을 갖는 문자(7E hex)가 있을 경우에는 제어탈출(control escape)문자(7D hex)가 이 문자열에 추가된다. 그래서 이것을 바이트 채우기 또는 문자채우기라고 한다.

다. 평가

이와 같이 정보기기의 표현계층에서 데이터를 원천부호화 할 때 부호에 비트 "1"이 연속하여 5개 이상 되도록 부호화하게 되면 데이터 링크계층에서 비트 또는 문자채우기를 통하여 이 부호를 다시 재부호화하는 일이 발생하게 된다. 이러한 채우기 동작이 빈번하게 발생되도록 원천 부호화할 경우에 이로 인한 전송지연 및 데이터의 증가로 전송율을 저하시키는 요인이 된다. 그러므로 문자를 원천 부호화할 때 연속 "1"의 비트가 4개 이상 되지 않도록 하는 것이 바람직하다.

3-2 뒤섞기(Scrambling)

가. 연속 "0" 비트열에 대한 뒤섞기

원천 부호가 일정이상의 연속 "0"비트열로 구성되어 있을 때, 이 "0"의 연속 비트열을 비연속 "0"의 비트열로 변경시킨다. 이 스크램블링 방식에는 크게 두 가지 국제 표준이 있다. 미국 및 일본 규격[7][8]과 ITU-T규격[9][10]이 이에 해당된다. 국내규격은 ITU-T의 규격을 준용하여 한국 정보 통신기술협회(TTA)에서 제정한 규격[17]이 있다. 본 논문에서는 TTA와 ITU-T규격을 중심으로 연구결과를 도출하였

다. 원천데이터를 교류적 전송 신호 성분으로 바꾸는 방법의 하나인 AMI에 대한 기술적 사항은 본 논문에서는 상세히 다루지 않았다.

나. 미국 및 일본방식

미국과 일본은 스크램블링방식으로 B8ZS(Bipolar with 8 zero substitution)을 사용한다. 이 규격은 연속 "0"의 비트가 8개 이상 입력될 때 8개의 "0"비트를 정해진 비트 열로 바꾸어 주는 방법이다. 바꾸어 주는 비트열은 000VB0VB가 된다. 이 방식은 AMI방식으로 인한 단점을 보완 하기 위한 것이다. 즉 연속된 "0"으로 인 한 동기를 잃어버리는 문제를 해결하기 위한 기술이다. V(Violation)비트는 AMI규칙을 따르지 않는 강제로 삽입 되는 준위로서 비로 이전 "1"의 극성과 같은 극성을 갖게 한다. B(Bipolar)는 AMI규칙을 적용한 준위이다. V와 B에 대한 극성의 결정은 연속되는 "0"의 바로 이전의 "1"의 비트에 대한 극성에 따라 결정된다.

다. ITU-T 및 국내규격

ITU-T와 우리나라는 스크램블링 방식으로 HDB3(High Density Bipolar 3zero)방식을 적용하고 있다. 이 방식은 원천 데이터로부터 연속"0"의 비트열이 4개 이상일 때 이 4개의 "0"비트를 정해진 비트열로 바꾸어 주는 방식이다. 이 방식도 AMI방식의 단점을 보완하기 위한 방법이다. 연속된 4개의 "0"비트 열은 000V 또는 B00V중 하나로 바뀐다[10]. V와 B에 대한 정의와 적용방법은 B8ZS의 경우와 동일하다. 두 개 중 하나를 결정하는 기준은 이전 비트 열 중에서 "0"이 연속인 비트가 정해진 비트 열로 대체된 후에 발생된 "1"의 개수에 의해 결정한다. "1"의 개수가 홀수인 경우에는 000V로 대체되고 "1"의 개수가 짝수일 때는 B00V로 대체된다.

즉 송신장치의 부호기에서는 "0"의 연속도 검색하여야 하며 정해진 비트 열로 "0"의 연속 비트열이 대체된 후의 "1"의 개수도 몇 개인지 계수하여 그 계수 정보를 실시간 업데이트하여 저장하고 있어야 한다. 수신기의 역부호기에서는 연속되어 입력되는 비트열

로부터 대체되어 있는 비트열을 찾아내어 원래 비트열로 복원시켜주어야 한다.

총186개로 나타났다.

IV. 문자 원천부호화 규칙

본 논문에서는 문자 원천 부호화 규칙을 논하는데 OSI(open system interoperability) 물리계층에서의 HDB-3 회선(line) 부호화 방식과 데이터링크 계층의 HDLC와 PPP 프로토콜 프레임의 플래그(flag) 필드를 기준으로 하였다. 이러한 조건하에서 문자 원천 부호화하는 규칙으로 다음과 같은 규칙을 적용하였다. 1개의 조합단위를 구성하는 3비트열, 4비트 열 또는 단위조합 비트열과 단위조합비트열 간의 조합에서 "0"의 연속과 "1"의 연속이 발생되지 않도록 한다. 여기서 1바이트는 행3 x 열4=7비트 또는 행4 x 열4=8비트(1바이트)로 구성된다. 단위조합간의 결합은 행4,4 x 열4,4=16비트(2바이트) 또는 행4,4,4 x 열4=16비트(2바이트)로 구성된다.

4-1 4비트 열 x 4비트 행 단위조합

표 1은 이러한 원칙을 적용한 4비트 x 4비트 단위 조합에 대한 각 비트열 별 조합이 가능한 것과 조합이 제한되는 조합조건을 보여 주는 것이다[12].

예를 들어 16진수(HEXA) 0의 경우에는 자체가 4개의 "0"의 연속으로 되어 있으므로 원칙적으로 조합에서 제외된다. 16진수 1의 경우에는 첫 번째 상위 비트열이 "0001"이므로 하위 비트열에 "1111" (HEXA; F)이 오지 않는 한 그 외의 비트열과 모두 조합이 가능하다. 이러한 조건으로 볼 때 16진수 F의 경우가 조합제한 개수가 8개로 가장 많고 16진수 8이 7개, 16진수 7이 4개, 16진수 4와 C가 3개, 16진수 3과 B가 2개 그리고 기타 16진수 1, 2, 5, 6, 9, A, D, E가 각각 16진수 1과 F와 조합이 제한되는 것으로 분석되었다. 이러한 방식으로 조합할 경우 조합이 가능한 총256개 중에서 70개가 하위비트열과 조합이 제한된다. 따라서 조합이 가능한 하위 비트열의 수는

표 1. 문자 부호화 규칙 ; 4 비트 x 4비트

Table 1. Characters coding Rule; 4-bit x 4-bit

HEXA	상위 비트열	하위 비트행	
		조합제한	조합가능
0	0000	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F	X
1	0001	0,F	1,2,3,4,5,6,7,8,9,A,B,C,D,E
2	0010	0,1	2,3,4,5,6,7,8,9,A,B,C,D,E,F
3	0011	0,E,F	1,2,3,4,5,6,7,8,9,A,B,C,D
4	0100	0,1,2,3	4,5,6,7,8,9,A,B,C,D,E,F
5	0101	0,F	1,2,3,4,5,6,7,8,9,A,B,C,D,E
6	0110	0,1	2,3,4,5,6,7,8,9,A,B,C,D,E,F
7	0111	0,C,D,E,F	1,2,3,4,5,6,7,8,9,A,B
8	1000	0,1,2,3,4,5,6,7	8,9,A,B,C,D,E,F
9	1001	0,F	1,2,3,4,5,6,7,8,9,A,B,C,D,E
A	1010	0,1	2,3,4,5,6,7,8,9,A,B,C,D,E,F
B	1011	0,E,F	1,2,3,4,5,6,7,8,9,A,B,C,D
C	1100	0,1,2,3	4,5,6,7,8,9,A,B,C,D,E,F
D	1101	0,F	1,2,3,4,5,6,7,8,9,A,B,C,D,E
E	1110	0,1	2,3,4,5,6,7,8,9,A,B,C,D,E,F
F	1111	0,8,9,A,B,C,D,E,F	1,2,3,4,5,6,7

*하위비트 행의 숫자는 16진수임

표 2는 4비트열 16진수 조합으로 구성된 8비트 바이트간의 조합규칙에 대한 것이다[12]. 첫 번째 출력된 바이트의 최상위 비트열과 두 번째 출력된 바이트의 최하위 비트열과 조합되는 것이다. 예를 들어 첫 번째 바이트의 최상위 비트열이 16진수 1인 경우 최상위 비트열은 0001이 된다. 두 번째 출력된 바이트의 최하위 비트열이 16진수 2인 경우 비트열이 0010이 된다. 따라서 두 바이트의 조합은 0010 0001이 된다. 결과적으로 두 바이트간의 조합이 "0"이 연속 4개로 이루어지게 된다. 그러므로 전체 비트열조합에 의한 바이트가 구성되면 각 바이트간의 결합에 대한 것을 고려하여야 한다. 표 2의 분석결과에 의하면 처

음 출력되는 바이트의 최상위 4개의 비트열이 16진수 1인 경우, 즉 “0001”비트열인 경우에 두 번째 출력되는 바이트의 최하위 4개의 비트열 16진수 중에서 0, 2, 4, 6, 8, 10, 12, 14, 15의 10개가 결합이 부적합한 것으로 나타났다. 처음 바이트의 최상위 비트열의 16진수가 2과 3인 경우, 두 번째 바이트의 최하위 비트열과 결합이 부적합한 16진수는 0, 4, 10, 14의 4개로 나타났다. 처음 바이트의 최상위 비트열의 16진수가 4, 5, 6 및 7인 경우, 두 번째 바이트의 최하위 비트열과 결합이 부적합한 16진수는 0, 8의 2개로 나타났다. 처음 바이트의 최상위 비트열의 16진수가 8~F인 경우, 두 번째 바이트의 최하위 비트열과 결합이 부적합한 16진수는 0, F의 2개로 나타났다.

표 2. 8비트간 조합규칙

Table 2. Composition rule between 8bits

16진수	첫 번째 바이트 최상위 비트열	두 번째 바이트 최하위 비트열	
		조합제한	조합가능
0	0000	모두	-
1	0001	0,2,4,6,8,10,12,14	1,3,5,7,9,11,13,15
2	0010	0,4,10,14	1,2,3,5,6,7,8,9,11,12,13,15
3	0011	0,4,10,14	1,2,3,5,6,7,8,9,11,12,13,16
4	0100	0,8	1,2,3,4,5,6,7,9,10,11,12,13,14,15
5	0101	0,8	1,2,3,4,5,6,7,9,10,11,12,13,14,15
6	0110	0,8	1,2,3,4,5,6,7,9,10,11,12,13,14,15
7	0111	0,8	1,2,3,4,5,6,7,9,10,11,12,13,14,15
8	1000	0,F	1,2,3,4,5,6,7,8,9,10,11,12,13,14
9	1001	0,F	1,2,3,4,5,6,7,8,9,10,11,12,13,14
A	1010	0,F	1,2,3,4,5,6,7,8,9,10,11,12,13,14
B	1011	0,F	1,2,3,4,5,6,7,8,9,10,11,12,13,14
C	1100	0,F	1,2,3,4,5,6,7,8,9,10,11,12,13,14
D	1101	0,F	1,2,3,4,5,6,7,8,9,10,11,12,13,14
E	1110	0,F	1,2,3,4,5,6,7,8,9,10,11,12,13,14
F	1111	0,F	1,2,3,4,5,6,7,8,9,10,11,12,13,14

4-2 3비트 열 x 4비트 행 단위조합

표 3은 이러한 원칙을 적용한 3비트 x 4비트 단위 조합에 대한 각 단위조합별 조합이 가능한 것과 조합이 제한되는 조합조건을 보여 주는 것이다[18]. 예를 들면 상위비트열 “000”은 하위 비트열 4개중에서 상위비트열의 최하위비트와 연결되는 최상위비트가 “0”에 해당되는 16진수 0~7까지의 비트열과는 조합이 제한된다. 이것은 4개의 “0”비트조합이 일어나는 것을 피하여야 하기 때문이다. 상위 비트열이 “111”인 경우에는 하위비트열 중에서 상위비트열과 조합되는 하위비트열의 최상위비트가 “1”이 되는 모든 하위 비트열을 피하도록 하여야 한다. 즉 상위비트열 “111”은 하위비트열의 최하위비트인 “1”과 상위비트열의 최상위비트와 차상위비트가 “1”이 되는 즉“11”이 연속되는 16진수 C~F와의 조합을 피하고 그 외의 숫자들과 조합되어야 한다.

표 3. 문자 2진 부호화 규칙; 3 비트행 x 4비트열
Table 3. Characters coding Rule ; 3-bit x 4-bit composition rule

3비트 열 (상위비트열)	4비트 행(하위 비트열)	
	조합제한	조합가능
000	0,1,2,3,4,5,6,7	8,9,A,B,C,D,E,F
001	0,F	1,2,3,4,5,6,7,8,9,A,B,C,D,E
010	0,1	2,3,4,5,6,7,8,9,A,B,C,D,E,F
011	0,E,F	1,2,3,4,5,6,7,8,9,A,B,C,D
100	0,1,2,3	4,5,6,7,8,9,A,B,C,D,E,F
101	0,F	1,2,3,4,5,6,7,8,9,A,B,C,D,E
110	0,1	2,3,4,5,6,7,8,9,A,B,C,D,E,F
111	0,C,D,E,F	1,2,3,4,5,6,7,8,9,A,B

* 3비트 열()내의 숫자는 10진수임. 4비트행의 숫자는 16진수임

표 4는 3x4비트열 조합으로 구성된 7비트 바이트 간의 조합규칙에 대한 것이다[18]. 첫 번째 출력된 바이트의 최상위 비트열과 두 번째 출력된 바이트의 최하위 비트열과 조합되는 것이다. 첫 번째 바이트의 최상위 비트열이 2진수 1인 경우 비트열은 001이 된다. 두 번째 출력된 바이트의 최상위 비트열이 16진수 4인 경우 비트열이 0100이 된다. 따라서 두 바이

트의 조합은 0100 001이 된다. 결과적으로 두 비트열 간의 조합이 “0”이 연속4개로 이루어지게 된다. 그러므로 전체 비트열조합에 의한 바이트가 구성되면 각 바이트간의 결합에 대한 것을 고려하여야 한다. 표 4의 분석결과에 의하면 처음 출력되는 바이트의 최상위 3개의 비트열이 2진수 0인 경우 두 번째 출력되는 바이트의 최하위 4개의 비트열 16진수 중에서 0, 2, 4, 6의 4개가 결합이 부적합한 것으로 나타났다. 처음 바이트의 최상위 비트열의 2진수가 1 경우, 두 번째 바이트의 최하위 비트 열과 결합이 부적합한 16진수는 0, 4, 8의 3개로 나타났다. 처음 바이트의 최상위 비트열의 2진수가 2, 3인 경우, 두 번째 바이트의 최하위 비트 열과 결합이 부적합한 16진수는 0, 8의 2개로 나타났다[18].

처음 바이트의 최상위 비트열의 2진수가 4, 5인 경우, 두 번째 바이트의 최하위 비트 열과 결합이 부적합한 16진수는 0, F의 2개로 나타났다. 마지막으로 처음 바이트의 최상위 비트열의 2진수가 7인 경우, 두 번째 바이트의 최하위 비트 열과 결합이 부적합한 16진수는 0, 3, 7, B, F의 5개로 나타났다.

표 4. 7비트-바이트간 조합규칙
Table 4. Composition rule between 7bits-Bytes

첫번째 바이트 상위 비트열 (3비트)	두번째 바이트 하위비트열 (4비트)	
	조합제한	조합가능
000	0,2,4,6	1,3,5,7,8,9,A,B,C,D, E,F
001	0,4,8	1,2,3,5,6,7,9,A,B,C,D ,E,F
010	0,8	1,2,3,4,5,6,7,9,A,B,C, D,E,F
011	0,8	1,2,3,4,5,6,7,9,A,B,C, D,E,F
100	0,F	1,2,3,4,5,6,7,8,9,A,B, C,D,E
101	0,F	1,2,3,4,5,6,7,8,9,A,B, C,D,E
110	0,7,F	1,2,3,4,5,6,8,9,A,B,C, D,E
111	0,3,7,B,F	1,2,4,5,6,8,9,A,C,D,E

V. 연구결과

본 논문은 컴퓨터 등 정보기기로부터 생성되는 문자부호를 전송로를 통하여 원거리에 전송할 때 전송 효율을 높일 수 있는 문자부호화 원칙을 제시하였다. 정보 기기 내에서 문자를 어떠한 규칙에 의해 전송하는가에 따라 데이터 전송에 대한 효율에 영향을 줄 수 있다.

현재 ITU-T와 국내의 물리계층에서 이루어지는 회선부호화(Line coding)방식은 HDB-3방식을 채택하고 있다. 이 방식은 연속되는 “0”의 비트가 4개이상 되지 않도록 하는 방식이다. 이 방식을 채택한 목적은 연속“0”비트로 인한 동기의 소실을 막기 위한 것이다. 따라서 문자를 “0”의 연속이 4개 이상 발생되지 않도록 부호화할 경우 전송효율을 상대적으로 증가시킬 수 있게 된다. 또한 데이터링크 계층에서는 HDLC와 PPP 프로토콜을 사용하고 있다. 이 두 개의 프로토콜의 프레임에는 프레임의 시작과 끝을 인식하기 위한 플래그 필드로 구성되어 있다. 이 프로토콜에서는 플래그 필드를 구성하는 비트열이 프레임 내의 다른 데이터비트열과 오인식되는 것을 방지하기 위해 데이터열에 “1”의 비트가 연속하여 5개 이상 발생될 경우 연속 “1”의 5개 비트다음에 인위적으로 “0”을 끼워넣거나 문자를 추가하는 방법을 사용하고 있다. 그러므로 문자의 원천부호가 연속 5개 “1”로 구성되지 않도록 한다면 비트나 문자의 끼워넣기로 인하여 발생하는 데이터의 전송효율을 상대적으로 증가시킬 수 있게 된다. 본 논문에서는 이점에 착안하여 전송빈도수가 높은 문자의 원천부호에 “0”의 연속이 4개 이상 그리고 “1”의 연속이 5개 이상 발생하지 않도록 하는 문자 원천 부호화 규칙을 제시하였다. 본 논문에서 제시한 원천부호화규칙을 적절히 활용할 경우, 데이터의 전송효율을 크게 제고시킬 것으로 기대된다. 향후 본 논문에 이어 추가로 연구하여야 사항은 다음과 같다. 첫째 본 논문에서 제시한 원천부호화 규칙을 적용하여 문자의 사용빈도에 따라 최적의 문자 원천부호화를 도출하는 것이다. 그리고 그 결과를 토대로 원천부호화 적용전과 후에 얻어지는 데이터 전송효율을 분석하는 것이다. 둘째 HDB-3 이외의 B8ZS 등 다른 회선부호방식에 대하여도 원천

부호화 규칙을 연구하는 것이다.

참 고 문 헌

- [1] RFC 20 ASCII format for Network Interchange October 1969(<http://tools.ietf.org/html/rfc20>)
- [2] American Standards Association, "American Code (July 6, 1999). for Information Interchange", *ASA X3.4- 1963*, 17 June 1963
- [3] American National Standards Institute, "American National Standard for Information Systems-Coded Character Sets 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII)", *ANSI X3.4-1986, Inc.*, 26 March 1986
- [4] <http://en.wikipedia.org/wiki/EBCDIC>
- [5] <http://en.wikipedia.org/wiki/Unicode>
- [6] ISO/IEC 13239:2002 Information technology-Telecommunications and information exchange between systems-High-level
- [7] <http://searchnetworking.techtarget.com/definition/B8ZS>
- [8] Behrouz A. Forouzan, "Data communications and Networking" *Fourth Edition. McGraw Hill Korea*, p116. May, 2007.
- [9] ITU-T Recommendation G.703, "Physical/ electrical characteristics of hierarchical digital interfaces", pp. 24-41, Oct., 1998.
- [10] Behrouz A. Forouzan, "Data communications and Networking" *Fourth Edition. McGraw Hill Korea*, p.117. May, 2007. data link control (HDLC) procedures
- [11] 산업자원부 기술표준원. "정보 교환용 부호계 (한글 및 한자).“ *KS X 1001 : 2004*, 2004. 12.. 28.
- [12] 홍완표, "데이터 전송 효율을 고려한 4비트행x4비트열 2 바이트 문자 부호화 규칙에 관한 연구", *한국향행학회 논문지, 제15권 제5호*, 2011년 10월
- [13] http://en.wikipedia.org/wiki/High-Level_Data_Link_Control
- [14] "Data Communication Lectures of Manfred Lindner-Part HDLC"http://en.wikipedia.org/wiki/High-Level_Data_Link_Control. Reference
- [15] Behrouz A. Forouzan, "Data communications and Networking" *Fourth Edition. McGraw Hill Korea*, p.339. May, 2007.

- [16] Behrouz A. Forouzan, "Data communications and Networking" *Fourth Edition. McGraw Hill Korea*, p.345. May, 2007.
- [17] TTA Standard, "Test Method for Telecommunication Terminal Equipment" *TTAS. Ko-05.0028/R1* pp306-451, Revised on 23 Dec., 2004
- [18] 홍완표, "데이터 전송 효율을 고려한 3x4비트 1바이트 문자부호화 규칙에 관한 연구" *한국전자통신학회 논문지 제6권 제4호*. 2011. 08. 12.

홍 완 표 (洪完杓)



- 1991년 서울과학기술대학교 전자공학과 졸업(공학사)
 1994년 연세대학교대학교 공학대학원 산업공학과 졸업(공학석사)
 1999년 광운대학교 대학원 전자공학과 졸업(공학박사)
 1990년 전기통신기술사합격
 1991년 정보통신부 5급특별채용고시 합격
 본부 통신정책실, 전파방송관리국, 정보화기획실
 1997년 삼성전자(주) 통신사업부 전송영업그룹장
 1999년 광운대학교 연구전담교수
 2000년 한국정보통신기술협회장
 2002년 한세대학교 IT학부 정보통신공학전공 교수
 한세대학교 정보통신연구소장
 관심분야 : 위성통신방송/문자코딩/통신정책/