

論 文

해밍코드를 이용한 효율적인 Hybrid ARQ 시스템의 성능분석

正會員 朴 性 卿* 正會員 金 信 始** 正會員 康 昌 彦**

Performance Analysis of the Hybrid ARQ System Using Hamming Codes

Seong Kyeong PARK*, Sin Ryeong KIM**, Chang Eon KANG** *Regular Members*

要 約 본 논문에서는 유한 버퍼용량을 갖는 SR (Selective-Repeat) ARQ (Automatic-Repeat-Request) 방식과 1개의 에러는 정정하고 2개의 에러를 검출하는(63, 56) 순환해밍 코드를 결합한 hybrid ARQ 방식에 대하여 연구하였다. 컴퓨터 시뮬레이션 결과 1개의 에러를 정정함으로써 전송효율이 증가됨을 알 수 있었으며, 2개의 에러를 검출하여 재전송에 의해서 뒤바뀐 코드워드의 순서를 바로잡아 수신자(user)에게 전달함을 보여주었다. 성능분석에서는 FEC (Forward-Error-Correction) 방식이나 이상적인 SR ARQ 방식보다 본 시스템의 전송효율과 신뢰도가 우수하며 특히, 채널에러율이 $10^{-2} \sim 10^{-3}$ 정도로 높은 경우 더욱 효율적임을 보여주었다.

ABSTI ACT In this paper, the hybrid ARQ scheme, which is incorporated the selective-repeat ARQ system with the finite receiver buffer and the single-error-correcting and double-error-detecting (63,56) cyclic Hamming code system, has been investigated.

As a result of simulation, the proposed hybrid ARQ scheme shows that the throughput efficiencies are improved by one error correction, and that the reversed codewords due to retransmission are delivered to the user in order by means of detecting two errors. The hybrid ARQ scheme significantly outperforms the FEC or the ideal selective-repeat ARQ system in the respect of throughput and reliability, especially when the channel error rate is approximately in the range from 10^{-2} to 10^{-3} .

I. 서 론

데이터통신시스템에서 채널상의 잡음등으로 인

하여 발생하는 에러를 해결하기 위한 에러제어방식으로는, 자주 발생하는 에러형태를 정정해서 재전송 횟수를 줄임으로써 전송효율(throughput)을 증가시키는 FEC(Forward Error Correction)방식과 자주 발생하지 않는 집단 에러형태를 검출해서 에러가 없는 데이터가 수신될때까지 재전송을 요구해서 신뢰도(reliability)를 증가시키는 ARQ(-

*陸軍通信將校

**延世大學校 工科大學 電子工學科

Dept. of Electronics, Yonsei University, Seoul 120-749, Korea.

論文番號 : 88-55(接受 1988. 11. 17)

Automatic Repeat Request)방식, 그리고 이 두방식을 결합한 hybrid ARQ방식이 있다. 본 연구에서는 FEC방식과 ARQ방식을 결합하여 성능(performance)을 향상시킨 hybrid ARQ방식에 대하여 연구한다. 1개의 에러는 정정하고 2개의 에러와 또는 에러형태가 1보다 크고 원시다항식으로 나누어지는 홀수개의 에러를 검출하는 (63,56)순환해밍 코드(cyclic Hamming codes)를 이용한 디코더를 버퍼용량 N인 SR ARQ방식과 결합시킴으로써 전송효율과 신뢰도를 향상시킨 hybrid ARQ시스템을 구성하였고 컴퓨터 시뮬레이션을 하여 성능 분석을 수행하였으며 또한 실험을 통하여 동작을 확인하였다.

II 에러 제어 방식

II-1-에러-정정, 2-에러-검출 순환해밍 코드(cyclic Hamming codes)

FEC방식의 한 종류로서 $p(x)$ 를 원시다항식이라 하면 생성다항식 $g(x)=(1+x) \cdot p(x)$ 로써 인코딩되고 디코더는 에러트래핑 디코더를 이용하여 구성된다^(1,3). (63, 56) 순환해밍 코드는 1개의 에러가 발생하면 에러트래핑 디코더를 이용하여 정정하고, 2개의 에러와 에러형태가 1보다 크고 원시다항식으로 나누어지는 홀수개의 에러를 검출할 수 있다.

II-2. ARQ(Automatic Repeat Request)방식

ARQ방식은 수신 코드워드에서 에러가 검출되었을때 송신단에서 코드워드를 재전송하는 방식에 따라 SW(Stop-and-Wait), GBN (Go-Back-N), SR(Selective-Repeat)ARQ방식으로 나뉘어진다⁽³⁾. 버퍼용량 N인 SR ARQ방식의 송신단은 수신단에서 오우버-플로우(overflow)가 발생하는것을 검출하여 잃어버린 코드워드를 재전송하고 수신단에서는 송신단에서 보내오는 코드워드를 받아서 에러검출을 시도하며, N개의 버퍼와 3N 순환범위의 계열번호(sequence number)를 사용하고 이때 같은 코드워드에 연속해서 두번 에러가 검출되면 오우버-플로우가 발생한다.

전송효율의 하한을 구해 보면 코드워드의 총비

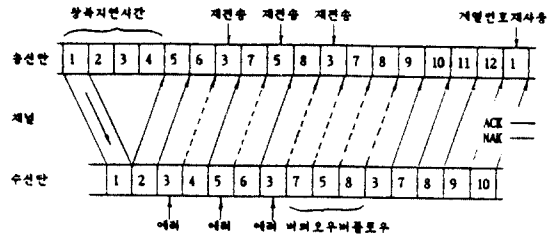


그림 1 버퍼용량 N인 SR ARQ(N=4) Selective-Repeat ARQ with a buffer size N(N=4)

트수를 n이라하고 한비트에 대한 에러확률을 e라 하면 하나의 코드워드가 정확히 전송 될 확률 $P=(1-e)^n$ 이며 하나의 코드워드가 한번, 두번, 세번의 재전송을 통하여 에러없이 전송 될 확률을 각각 $A=1-(1-P)^2$, $B=1-(1-P)^3$, $C=1-(1-P)^4$ 라하면 식(1)과 같다.

$$\eta_{out} \geq \frac{r_0}{(r_0 + r_1 + r_2 \cdot N)} \quad (1)$$

여기서

$$\left. \begin{aligned} r_0 &= \frac{B \cdot (1 - B \cdot C^{N-1})}{(1 - C)} \\ r_1 &= P^2 \cdot \{A^{N-2} + (1 - P) \cdot B^{N-2} + (1 - P)^2 \cdot C^{N-2}\} \\ r_2 &= 3 - P^2 \cdot A^{N-2} - A^2 \cdot B^{N-2} - B^2 \cdot C^{N-2} \end{aligned} \right\} \quad (2)$$

이다.

II-3 hybrid ARQ 방식

FEC방식은 전송효율은 좋으나 신뢰도가 떨어지고, ARQ방식은 전송효율은 떨어지나 신뢰도가 좋다는 장단점을 갖고 있으므로 이 두방식을 결합하면 각각의 단점을 극복할 수 있는데 이것이 hybrid ARQ방식이다^(6,7).

이 방식은 에러의 검출과 정정을 동시에 수행하는 코드를 사용하며 수신된 코드워드에서 에러가 검출되면 수신단은 먼저 에러정정을 시도하여 에

러의 갯수가 에러정정 능력 범위이내라면 에러가 정정되어 수신자에게 전달되거나 수신버퍼에 저장되며, 에러정정능력 범위를 벗어나면 수신단은 NAK를 보내서 재전송을 요구한다.

III. hybrid ARQ방식의 송수신단 설계

III-1 송신단 설계

송신단은 (63,56)순환해밍 코드의 원시다항식 $p(x)=1+x+x^6$ 라하면 생성다항식 $g(x)=(1+x) \cdot p(x)=1+x^2+x^6+x^7$ 에 의해 코드워드를 발생시키는 인코더와 왕복지연 시간동안 코드워드를 기억시켜 놓은 재전송버퍼, 수신단에서 오는 응답신호를 받아서 재전송여부를 판단하는 제어기, 그리고 카운터로 구성된다.

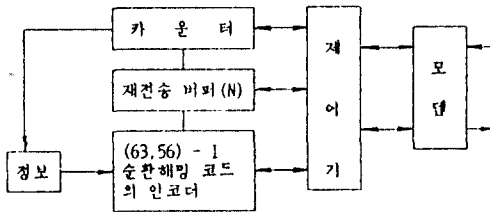


그림 2 송신단 구성도
Blockdiagram of the hybrid ARQ transmitter

정보는 코드워드가 되어 전송과 동시에 재전송 버퍼에 저장된다. 왕복지연 시간이 지난후 응답신호가 도착하는데 제어기에서 ACK인 경우는 새로운 코드워드를 전송하고 NAK인 경우는 재전송을 결정해주며, 이때 x 가 재전송버퍼의 가장 먼저 NAK된 계열번호이고 y 가 현재의 계열번호라면 전방거리 $F_t=y-x(\text{mod } 3N)$ 이 되고 이 전방거리와 응답신호에 따라 전송할 코드워드를 결정한다.

송신 시스템은 계산된 전방거리와 응답신호에 따라서 다음과 같이 동작한다.

1. ACK이고 $F_t < N$ 이면 (63,56)-1 순환해밍 코드의 인코더에 의해 생성된 첫번째 코드워드를 전송하고 재전송버퍼에 저장한다.
2. NAK 또는 unACK이고 $F_t < N$ 이면 재전송 버퍼에 있는 코드워드를 재전송 한다.
3. NAK 또는 unACK이고 $F_t = 0$ 이면 재전송버퍼에 있는 $F_t \geq N$ 인 코드워드를 (63,56)-1

순환해밍 코드의 인코더의 위치로 옮긴다.

이때, 수신단에서 버퍼-오우버-플로우가 발생했다.

4. $F_t \geq N$ 이면 인코더의 첫번째 코드워드를 전송한다.

III-2 수신단 설계

hybrid ARQ 시스템의 수신단은 송신단에서 보내오는 코드워드에서 에러를 검출하는 (63,56) 순환해밍 코드의 디코더와, 에러가 검출 되었을 때 코드워드의 순서를 바로 잡기위한 N개의 수신버퍼, 응답신호 발생기, 수신 코드워드를 수신자에게 전달할 것인지 재전송을 요구할 것인지를 판단하는 제어기와 발생한 신호를 채널로 보내는 모뎀 그리고 수신된 데이터를 수신자에게 전달하는 인터페이스 역할을 하는 게이트로 구성된다.

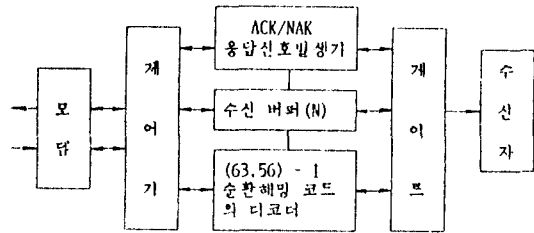


그림 3 수신단 구성도
Blockdiagram of the hybrid ARQ receiver

디코더에 의해서 검출된 에러가 없거나 또는 1개의 에러가 검출되면 에러트래핑 디코더를 통하여 에러가 정정된 후 수신자에게 전달하고, 2개이상 검출된 에러는 재전송을 통하여 NAK를 발생하여 재전송을 요구한 후 수신버퍼에 저장한다. 이때 코드워드를 회복하는 과정은, 버퍼가 비어있는 정규상태에서 y 가 수신단에 최후로 수신되어 수신자에게 전달된 계열번호이고 X 가 현재의 계열번호라면 전방거리 $F_t=x-y(\text{mod } 3N)$ 과, 수신버퍼에 코드워드가 저장되어있는 차단상태에서 y 가 NAK를 발생한 최초의 계열번호라면 전방거리 $L_t=x-y(\text{mod } 3N)$ 이 되고, z 가 버퍼내에 최후로 저장된 계열번호라면 후방거리 $L_b=z-x(\text{mod } 3N)$ 에 따라서 수신이 동작한다

hybrid ARQ 시스템의 수신단은 계산된 전방거리와 후방거리 그리고 수신 코드워드의 에러 유무에 따라 다음과 같이 동작한다.

먼저 정규 상태에서는

1. $F_r=1$ 이고 에러가 없으면 코드워드를 수신자에게 전달한 후 ACK를 전송하고, 에러가 검출되면 코드워드를 제거하고 NAK를 전송한 후 수신 버퍼에 공간을 확보하고 차단상태로 전환한다.
2. $1 < F_r \leq N$ 이고 에러가 검출되면 수신 버퍼에 공간을 확보하고, 에러가 없으면 코드워드를 수신 버퍼에 저장하고 차단상태로 전환한다.
3. $F_r > N$ 이면 송신단이 불필요한 재전송을 한 경우이므로 ACK를 전송하고 정규상태를 지속한다.

그리고 차단상태에서는

1. $L_r < N$ 이고 에러가 없으면 코드워드를 수신 버퍼에 저장한 후 ACK를 전송하고, 에러가 검출되면 수신 버퍼에 공간을 확보하고 NAK를 전송한다
2. $L_r \geq N$ 이고 $L_o < 2N$ 이면 ACK를 전송하고, $L_o \geq 2N$ 이면 버퍼-오우버-플로어가 발생했으므로 NAK를 전송하여 재전송을 받는다.
3. $L_r=0$ 이고 에러가 없으면 모든 에러없는 코드워드를 수신자에게 전달한다. 그후, 수신 버퍼가 비어 있으면 정규상태로 복귀하고 아니면 차단상태를 지속한다.

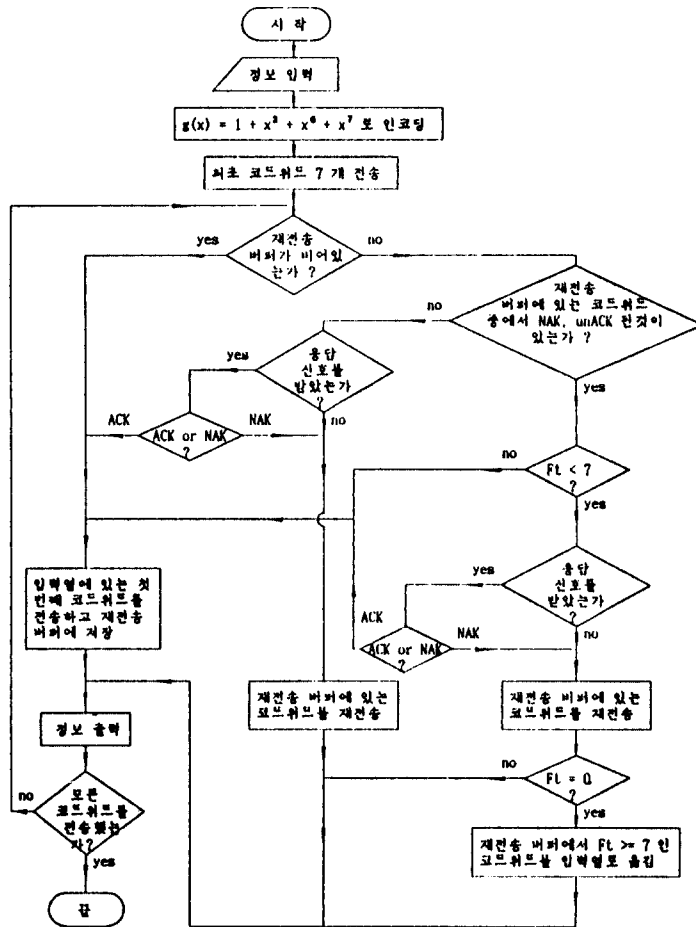


그림 4 송신단의 전송 및 재전송 과정
Transmission and retransmission procedure of the transmitter

IV. 시스템의 성능 분석

설계한 송수신단의 전향채널은 비트에러율 e를 갖는 랜덤에러채널이고, 케환채널은 응답신호가 에러없이 수신되는 무잡음채널이며, 계열번호를 나타내는 디지털에는 에러가 없다고 가정하여 왕복지연 시간 N=7인 경우에 대하여 컴퓨터 시뮬레이션을 하였다.

IV-1 송신단 시뮬레이션

송신단의 흐름도는 그림 4에 나타냈으며 ACK는 송신 코드워드가 에러없이 수신되었다는 응답 신호이며, NAK는 송신코드워드에 에러가 포함되어 수신되었다는 응답신호이고, unACK는 제한 채널에서의 응답신호 분실여부 상태로서 응답신서 응답신호가 제한채널에서 분실되어 수신하지

표 1 시뮬레이션 결과(송신단의 전송되는 코드워드)
Simulation results of the transmission procedure

un ACK 개별 ACK NAK 번호	전송되는 코드워드
1	0000111000001001111100011001000111011011110010110
2	0001101010010010101011111101010110101111111000110110101
3	000100010100101011110001010111000101111000111001101000110010
4	00011100111000000100101010110001111011110011001001111001011
5	001000010101000001010000111011100100001101101010010101110001
6	001010110110101110011111100001101000000010010101101100011001
7	001100011001001101000011111111111010100101110110111011010110
8	00110010010000111110110000001101100100111011100101001010110011
9	01000101111110000010010011010001110011010010010011001101101
10	00100010110010101110001010111000100111100011110011001001001010
11	0101101000011110001010111001110101001010101000111001101110010001
12	001010110110101100111111010000110100100000000100101011000110001
13	0101101111111010000010010100111000110001001110111001001010100100
14	01100101010110101010101010110110011011110010000010101011000100
15	011001000100001110010000001000010010101011100101110100010001010
16	0111111010101111111010100101100100011001011001011101011101011
17	10000001000011110001011101010010110010011001011001011101011
18	001010110110111100111110100011001001001001010101010000110000101
19	011001010110101001001101100110010011001100110010010010010000
20	01101000011001000001001010000110010011001011100100000110000
21	10100110001010110111101100100101110000110111101100010111011
22	0000011011010010011110010111000000011101010111010111000010
23	000100011010111110000011101001110001100101010000001011101011
24	00110101111100011000101110100101110010001110111001000111000101
25	00101000110001011100000100001000010000110000110000110100001
26	00101000110001011100000100001000010000110000110000110100001
27	001100011001001110011100000110101011100100110010111010000001
28	000101110101010101011110111000001110101100110010111010110000
29	000101110101010101011110111000001110101100110010111010110000
30	000101110101010101011110111000001110101100110010111010110000
31	000101110101010101011110111000001110101100110010111010110000
32	000101110101010101011110111000001110101100110010111010110000
33	000101110101010101011110111000001110101100110010111010110000
34	000101110101010101011110111000001110101100110010111010110000
35	000101110101010101011110111000001110101100110010111010110000
36	000101110101010101011110111000001110101100110010111010110000
37	000101110101010101011110111000001110101100110010111010110000
38	000101110101010101011110111000001110101100110010111010110000
39	000101110101010101011110111000001110101100110010111010110000
40	000101110101010101011110111000001110101100110010111010110000
41	000101110101010101011110111000001110101100110010111010110000
42	000101110101010101011110111000001110101100110010111010110000
43	000101110101010101011110111000001110101100110010111010110000
44	000101110101010101011110111000001110101100110010111010110000
45	000101110101010101011110111000001110101100110010111010110000
46	000101110101010101011110111000001110101100110010111010110000
47	000101110101010101011110111000001110101100110010111010110000
48	000101110101010101011110111000001110101100110010111010110000
49	000101110101010101011110111000001110101100110010111010110000
50	000101110101010101011110111000001110101100110010111010110000
51	000101110101010101011110111000001110101100110010111010110000
52	000101110101010101011110111000001110101100110010111010110000
53	000101110101010101011110111000001110101100110010111010110000
54	000101110101010101011110111000001110101100110010111010110000
55	000101110101010101011110111000001110101100110010111010110000
56	000101110101010101011110111000001110101100110010111010110000
57	000101110101010101011110111000001110101100110010111010110000
58	000101110101010101011110111000001110101100110010111010110000
59	000101110101010101011110111000001110101100110010111010110000
60	000101110101010101011110111000001110101100110010111010110000
61	000101110101010101011110111000001110101100110010111010110000
62	000101110101010101011110111000001110101100110010111010110000
63	000101110101010101011110111000001110101100110010111010110000
64	000101110101010101011110111000001110101100110010111010110000
65	000101110101010101011110111000001110101100110010111010110000
66	000101110101010101011110111000001110101100110010111010110000
67	000101110101010101011110111000001110101100110010111010110000
68	000101110101010101011110111000001110101100110010111010110000
69	000101110101010101011110111000001110101100110010111010110000
70	000101110101010101011110111000001110101100110010111010110000
71	000101110101010101011110111000001110101100110010111010110000
72	000101110101010101011110111000001110101100110010111010110000
73	000101110101010101011110111000001110101100110010111010110000
74	000101110101010101011110111000001110101100110010111010110000
75	000101110101010101011110111000001110101100110010111010110000
76	000101110101010101011110111000001110101100110010111010110000
77	000101110101010101011110111000001110101100110010111010110000
78	000101110101010101011110111000001110101100110010111010110000
79	000101110101010101011110111000001110101100110010111010110000
80	000101110101010101011110111000001110101100110010111010110000
81	000101110101010101011110111000001110101100110010111010110000
82	000101110101010101011110111000001110101100110010111010110000
83	000101110101010101011110111000001110101100110010111010110000
84	000101110101010101011110111000001110101100110010111010110000
85	000101110101010101011110111000001110101100110010111010110000
86	000101110101010101011110111000001110101100110010111010110000
87	000101110101010101011110111000001110101100110010111010110000
88	000101110101010101011110111000001110101100110010111010110000
89	000101110101010101011110111000001110101100110010111010110000
90	000101110101010101011110111000001110101100110010111010110000
91	000101110101010101011110111000001110101100110010111010110000
92	000101110101010101011110111000001110101100110010111010110000
93	000101110101010101011110111000001110101100110010111010110000
94	000101110101010101011110111000001110101100110010111010110000
95	000101110101010101011110111000001110101100110010111010110000
96	000101110101010101011110111000001110101100110010111010110000
97	000101110101010101011110111000001110101100110010111010110000
98	000101110101010101011110111000001110101100110010111010110000
99	000101110101010101011110111000001110101100110010111010110000
100	000101110101010101011110111000001110101100110010111010110000

못한 경우로 이 경우도 NAK로 간주한다. 시뮬레이션 결과인 표 1에서 처음 5 비트는 정보를 나타내고 다음 7 비트는 패리티검출디지털이다. 그리고 unACK의 '0'은 응답신호가 분실되어 송신단에 도착하지 않은 상태로 NAK와 같이 동작하며, '1'은 응답신호가 정확히 수신되었을 때를 나타낸다. 또한 ACKNAK에서 ACK는 '1'로, NAK는 '0'으로 나타냈는데, 송신단이 ACK로 간주할때는 unACK와 ACKNAK가 동시에 '1'인 경우 뿐이며 그외는 모두 NAK로 간주한다. 또한 계열번호는 2진수를 10진수로 나타낸다.

IV-2 수신단 시뮬레이션

수신단은 수신버퍼가 비어있는 정규상태와 수신버퍼에 코드워드가 저장되거나 공간을 확보해 놓은 차단상태로 분리된다.

그림 5은 정규상태 흐름도를, 그림 6는 차단상태 흐름도를 나타내었다. 정규상태에서는 에러가 순환해밍 코드의 디코더 의해서 정정할 수 없을 만큼 검출되거나 계열번호의 순서가 뒤바뀌면 차단상태로 동작전환하는데 이때는 전방거리, 후방거리, 에러유무에 따라 동작하며, 이때 같은 코드워드에서 연속해서 두번 이상 정정할 수 없는 에러가 발생하면 오우버-플로우가 발생한다.

시뮬레이션 결과인 표 2에서 X는 송신단에서 전송한 코드워드, E는 채널상의 에러형태, R은 채널을 통과한 수신 코드워드, C는 1개의 에러가 발생하였을때 에러트래핑 디코더를 이용하여 정정된 코드워드이며 앞에있는 숫자는 계열번호를 10진수로 표시한 것이고 끝에있는 숫자는 ACK를 '1'로, NAK를 '0'으로 표시했다. 결과에서 보면 1개의 에러가 발생했을때는 코드워드를 정정하여 수신자에게 전달한 후 ACK전송하고, 2개 검출된 에러는 수신버퍼에 저장한 후 NAK전송해서 재전송을 요구함을 알 수 있다.

IV-3 성능분석

P₀를 ARQ방식에서 코드워드에 에러가 없을 확률이라 하고, P₁를 hybrid ARQ방식에서 에러가

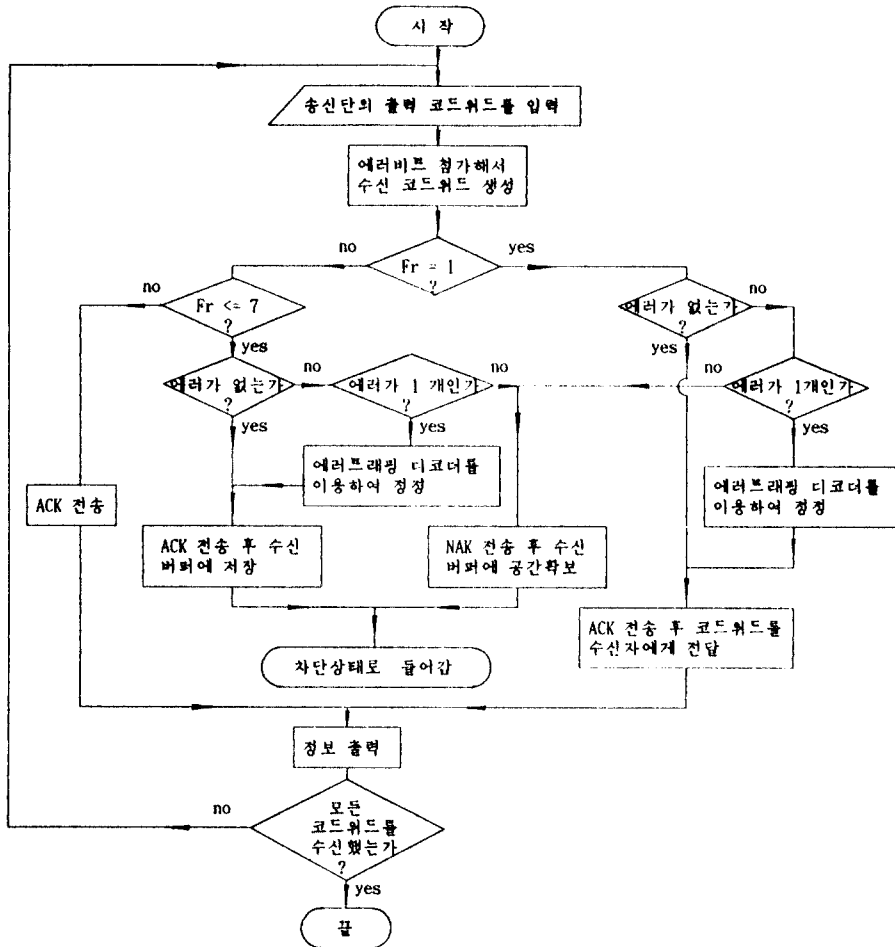


그림 5 수신단의 정규상태 동작
Normal-state operation of the receiver

없을 확률이라 하자. 그리고 P_{nd} 를 hybrid ARQ 방식에서 에러검출에 실패한 확률, P_{nc} 를 hybrid ARQ방식에서 에러정정에 실패한 확률, 그리고 P_e 를 코딩시스템을 사용하지 않을때 발생하는 에러 확률이라고 한다면 각각을 다음과 같이 나타낼 수 있다.

$$P_0 = (1 - e)^n \quad (3)$$

$$P_1 = \sum_{i=0}^1 \binom{n}{i} \cdot e^i \cdot (1 - e)^{n-i} \quad (4)$$

$$P_{nd} = \sum_{i=3}^n \binom{n}{i} \cdot e^i \cdot (1 - e)^{n-i} \quad (5)$$

$$P_{nc} = \sum_{i=2}^n \binom{n}{i} \cdot e^i \cdot (1 - e)^{n-i} \quad (6)$$

$$P_e = \sum_{i=1}^n \binom{n}{i} \cdot e^i \cdot (1 - e)^{n-i} \quad (7)$$

그림 7은 본 논문에서 시뮬레이션 한 $n=63$, $N=7$ 인 경우와 $n=512$, $N=128$ 인 경우의 전송효율을 나타내었다. 그림에서 보면 hybrid ARQ방식의 전송효율이 이상적 SR ARQ방식보다 우수함을 알 수 있으며, 특히, 왕복지연 시간이 길어짐에 따라 그 차이는 더욱 커져 채널에러율이 10^{-3} 인 경우에 이상적 SR ARQ방식과 버퍼용량 N인 SR ARQ방.

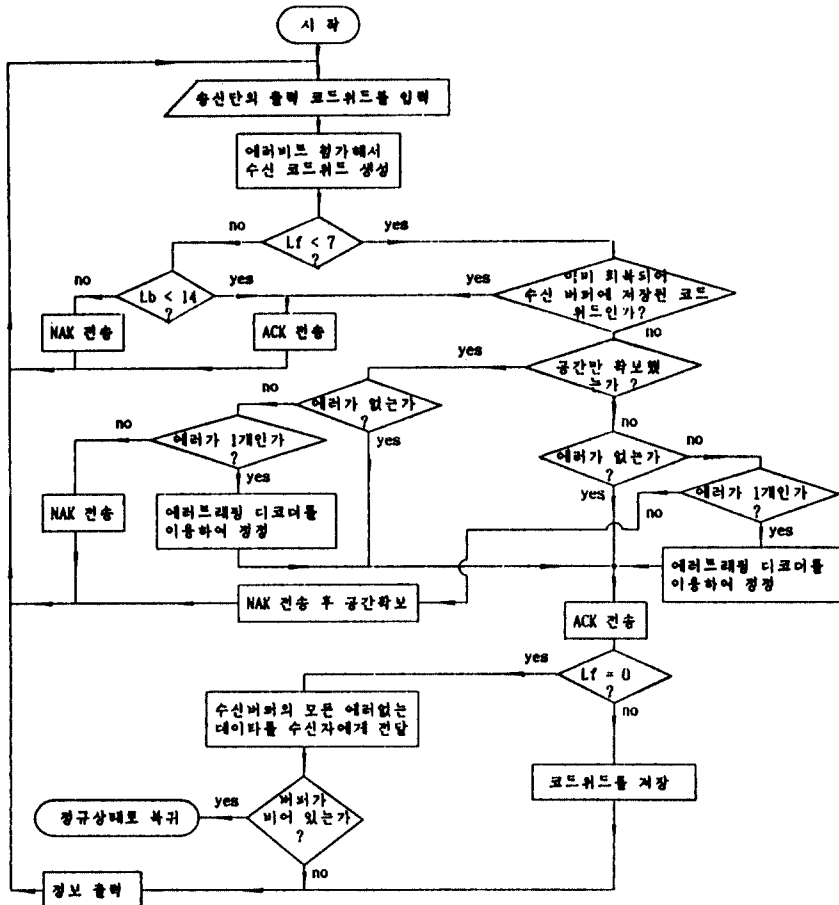


그림 6 수신단의 차단상태 동작
Blocked-state operation of the receiver

표 2 시뮬레이션 결과(수신과정)

Simulation results of the receive procedure

X: '1'	000001110000001000111110001101000110010001011000111101110111001010
E:	00
R:	000001110000001000111110001101000110010001011000111101110111001010 '1'
X: '2'	00001101010010010011010101110111010101010101011111110001101101001
E:	00001010101010010011010101110111010101010101011111110001101101001
R:	0000110101011001001101010111101110101010101011111110001101101001
C:	0000110101010010011010101111011101010101010111111110001101101001 '1'
X: '3'	00010001011001010111000010101110001001111000111001110100011001010
E:	00
R:	00010001011001010111000010101110001001111000111001110100011001010 '0'
X: '4'	00011110011110010000100101010110110001111011110011001001111001001
E:	00
R:	000111100111100100001001010101110001111011110011001001111001001
C:	0001111001111001000010010101011010001111011110011001001111001001 '1'
X: '5'	001000001010100000101000001101110100010000110110101000101011100001
E:	00
R:	001000001010100000101000001101110100010000110110001000101011100001
C:	0010000010101000001010000011011101000100001101101000101011100001 '1'
X: '6'	001010110110101110011111101000011010010000001000101011000110001
E:	0100
R:	001010101101011100111111010001101001000000100011011000110001 '0'
X: '7'	00110100011001001000001111101111110101000111011010111011010110
E:	00
R:	00110100011001001000001111101111110101000111011010111011010110 '1'

X: '8'	001110010100100011111011100000011011001001110110010100101010110011
E:	00
R:	0011100101001000111110111000000110110010011101110010100111010110011
C:	0011100101001000111110111000000110110010011101110010100101010110011 '1'
X: '9'	01000101111110000010010011010001110011101001110101001100011011101
E:	00
R:	0100010111111000001001001101000111001110101001110101001100011011101
C:	0100010111111000001001001101000111001110101001110101001100011011101 '1'
X: '3'	00010001011001010111000010101110001001111000111001110100011001010
E:	00
R:	00010001011001010111000010101110001001111000111001110100011001010 '1'
X: '10'	0100110000011101000101011100111010000111110010000101000111010101
E:	00
R:	0100110000011101000101011100111010000111110010000101000111010101
C:	01001100000111010000101011100111010000111110010000101000111010101 '1'
X: '11'	010101101100010100011101100101000111011000011100111011110010001
E:	00
R:	010101101100010100011101100101000111011000011100111011110010001
C:	010101101100010100011101100101000111011000011100111011110010001 '1'
X: '6'	00101011011010111001111110100001101001000000100010101100110001
E:	0100
R:	001010100110101110011111101000011010010000001000110111000110001 '0'
X: '12'	01011011111101000000100100001110011100100111011110010010110100100
E:	00
R:	01011011111101000000100100001110011100100111011110010010110100100
C:	01011011111101000000100100001110011100100111011110010010110100100 '1'

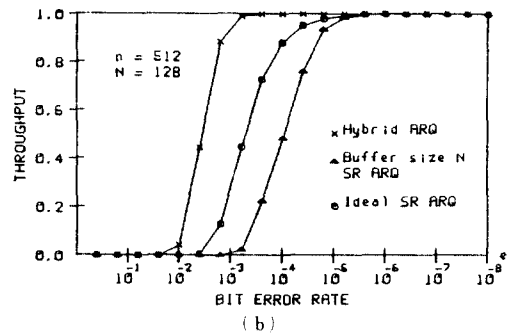
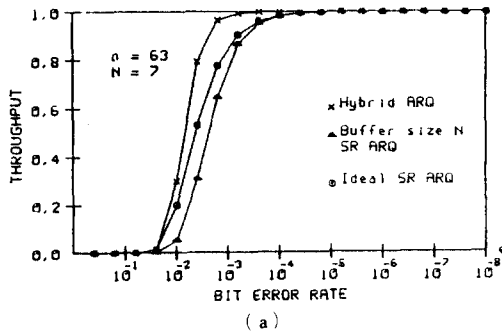


그림 7 전송효율 비교

Comparison of the throughput efficiencies

((a) $n=63, N=7$ (b) $n=512, N=128$)

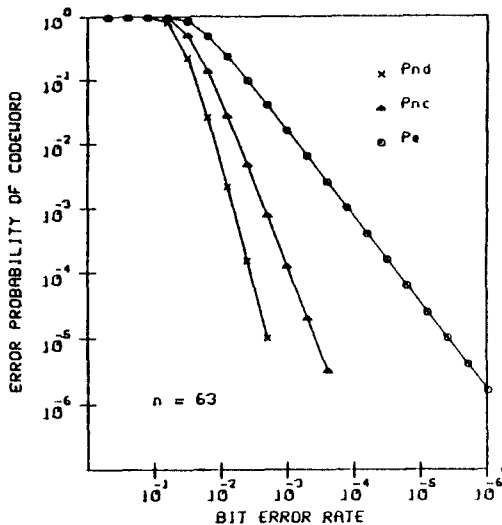


그림 8 신뢰도 비교

Comparison of the reliability efficiencies

식은 사용이 불가능할 정도로 떨어지나 hybrid ARQ 방식은 사용가능한 전송효율을 유지함을 알 수 있다.

또한, 신뢰도는 그림 8에 $n=63$ 인 경우에 대하여 나타내었는데 코딩시스템을 사용하지 않을 경우의 에러발생확률이 가장 많고, FEC방식에서는 에러 정정을 하므로 에러발생확률이 감소함을 알 수 있으며, hybrid ARQ방식은 에러정정 뿐 아니라 정정할 수 없는 에러도 재전송을 통하여 에러를 모두 회복하므로 에러 발생확률이 더욱 감소함을 알 수 있다.

V. 실험 및 결과 고찰

본 연구에서 제시한 시스템을 구현하기 위하여 송수신단의 환경을 변경하여 실험을 실시하였다. 그림 9에서 PC-I, PC-II는 각각 지구국과 위성국을 대체한 것이며 전송채널로는 기존의 위성채널대신에 RS-232C를 이용하였고, hybrid ARQ시스템의 제어기는 지구국과 위성국의 transceiver에 해당하는 것으로 통신환경을 변경하였다. 그리고 사용하는 전송 프로토콜로는 HDLC 프로토콜을 이용하였고 데이터전송속도는 $9600[b/s]$, 재전송 타임아웃은 $1023[ms]$, 윈도우크기는 7, 프레임의 길이는 최대 $512[bytes]$ 로 하였다. 본 연구에서 수행한 실험 구성도를 그림10에 나타냈다.

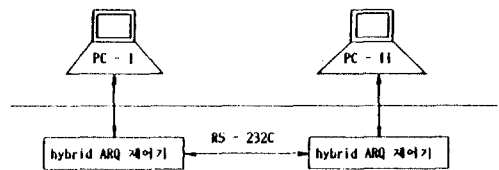


그림 9 송수신단 구성도

Blockdiagram of the transmitter and the receiver

실험과정은 우선 hybrid ARQ 시스템의 제어기를 setup하고 PC 간을 linkup 시킨다음 전송측은 전송 mode로, 수신측은 수신 mode로 위치시킨다. $9600(b/s)$ 의 정상적인 데이터전송속도에서는 전공하고자하는 데이터가 hybrid ARQ 시스템의 제어기에 의해서 프레임으로 변환.

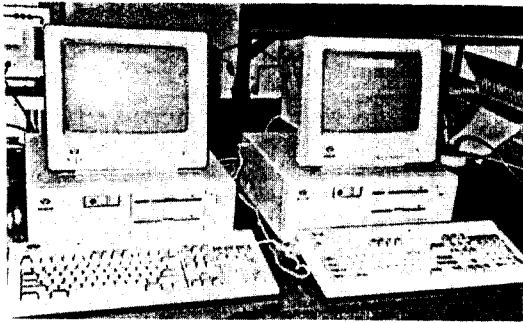


그림10 실험 구성도
Experimental construction

되어 거의 에러없이 전송되었음을 확인하였으나, 전송속도를 19200(b/s) 이상으로 hybrid ARQ 시스템의 제어를 동작시켰을 때는 에러가 발생하는데, 이 경우 동일한 화일을 전송하는데 걸리는 시간이 정상적인 상태보다도 길게 걸리는 것을 확인하였다. 이것은 에러에 의해서 프레임의 재전송하는데 걸리는 시간이 첨가되었기 때문이다.

VI. 결 론

본 논문에서 hybrid ARQ방식으로는 1개의 에러는 정정하고 2개의 에러와 에러형태가 1보다 크고 원시다항식으로 나누어지는 홀수개의 에러를 검출하는 (63,56)순환해밍 코드를 사용하고, ARQ방식으로는 버퍼용량 N을 갖는 SR ARQ방식과 결합하여 설계하였고 시스템의 성능 또한 분석하였으며 실험을 통하여 확인하였다. 컴퓨터 시뮬레이션 결과 1개의 에러는 정정하여 전송효율을 증가시켰으며 특히 10^{-3} 이상에서 효율적이며, 또한 2개의 에러와 에러 형태가 1보다 크고 원시다항식으로 나누어지는 홀수개의 에러를 검출하여 재전송을 통하여 신뢰도를 증가시켰는데 특히 10^{-2} 이상에서 효율적임을 알 수 있었다.

앞으로는 수신버퍼용량이 왕복지연 시간N보다 작을 경우와, 송신단에서 전송하는 코드워드의 계

열번호에도 에러가 발생할 경우에 전송효율을 높이기 위한 연구와 또한 응답신호가 폐환채널에서도 잡음이 발생할 경우와, 같은 성능을 유지하면서도 보다 간단한 방식에 대한 연구가 지속적으로 실시되어야 할 것이다.

參 考 文 獻

1. S. Lin, *An Introduction to Error-Correcting Codes*, Prentice-Hall, 1970.
2. W.W. Peterson and E.J. Weldon, *Error Correcting Codes*, 2nd ed., MIT Press, 1972.
3. S. Lin, *Error Control Coding*, Prentice-Hall, 1983.
4. M.J. Miller and S. Lin, "The analysis of some selective-repeat ARQ schemes with finite receiver" *IEEE Trans. Commun.*, COM-29, pp.1307-1315, Sept. 1981.
5. P.S. Yu and S. Lin, "An efficient selective-repeat ARQ scheme satellite channels and its throughput analysis", *IEEE Trans. Commun.*, COM-29, pp.353-363, March 1981.
6. S. Lin and P.S. Yu, "A hybrid ARQ scheme with parity retransmission for error control of satellite channel", *IEEE Trans. Commun.*, COM-30, no.7, pp.1701-1719, July 1982.
7. Y.M. Wang and S. Lin, "A modified selective-repeat type-II hybrid ARQ system and its performance analysis", *IEEE Trans. COM-31*, no.5, pp. 593-608, May 1983.
8. E.J. Weldon, Jr., "An improved selective-repeat ARQ strategy", *IEEE Trans. Commun. COM-30*, no.3, pp.480-486, March 1982.
9. A.R.K. Sastry, "Performance of hybrid error control schemes on satellite channel", *IEEE Trans. Commun.*, COM-23, pp.689-694, July 1975.



朴性卿(Seong Kyeong PARK) 正會員
1962年8月10日生
1981年3月~1985年2月: 金烏工科大学
電子工學科 卒業(工學士)
1987年3月~1989年2月: 延世大學校大
學院電子工學科卒業(工學
碩士)
1985年3月~現在: 陸軍通信將校



金信諤(Sin Ryeong KIM) 正會員
1960年12月12日生
1983年2月: 慶北大學校電子工學科卒業
(工學士)
1985年2月: 延世大學校 大學院 電子工
學科 卒業(工學碩士)
1985年3月~現在: 延世大學校 大學院
電子工學科 博士課程



康昌彦(Chang Eon KANG) 正會員
1938年8月26日生
1960年: 延世大學校電氣工學科(工學士)
1965年: 延世大學校大學院電氣工學科
(工學碩士)
1969年: 美國미시간주립大學校大學院電
氣工學科(工學碩士)
1973年: 美國미시간주립大學校大學院電
氣工學科(工學博士)

1967年~1973年: 美國미시간주립大學校工業研究所先任研究員

1973年~1981年: 美國노턴일리노이대학교電氣工學科助教授,
副教授

1982年~現在: 延世大學校電子工學科 教授

1987年~現在: 本學會 副會長