

CDMA/TDD 시스템에서 터보 부호를 사용한 H-ARQ 기법 연구

H-ARQ Scheme Using Turbo Code in CDMA/TDD Systems

조성철(S.C. Cho)	TDD전송기술연구팀 연구원
조대순(D.S. Cho)	TDD전송기술연구팀 연구원
이유로(Y.R. Lee)	TDD전송기술연구팀 연구원
정인철(I.C. Jeong)	TDD전송기술연구팀 선임연구원
어윤(Y. Uh)	TDD전송기술연구팀 초빙연구원
권동승(D.S. Kwon)	TDD전송기술연구팀 책임연구원, 팀장

IMT-2000에서 제공하고자 하는 인터넷 서비스나 멀티미디어 서비스를 위하여 고속 패킷 전송이 이루어져야 한다. 이러한 고속 패킷 서비스의 경우 불연속적인 데이터의 전송 및 비대칭적 데이터 트래픽의 특성으로 인하여 최근 CDMA/TDD 시스템에 대한 연구가 관심의 초점이 되고 있다. 따라서 본 고에서는 TDD 방식을 사용한 시스템의 특성을 고려하여, 전송 프레임의 구성 방식, 오류 발생 프레임의 재전송 방식, 채널 부호화 방식 등에 대하여 고찰하도록 한다.

I. 서론

음성위주의 서비스에서는 UE(User Equipment)에서 Node-B(기지국)로 향하는 업링크나 Node-B에서 UE로 향하는 다운링크를 통하여 전송되는 데이터량의 차이가 별로 없으나, 멀티미디어 서비스의 경우에는 서비스 종류에 따라 업링크와 다운링크의 전송 데이터량에 있어서 차이가 크게 나타날 수 있다. 업링크에서는 적은 양의 데이터가 이동하지만, 다운링크는 실시간 동영상의 전송, 파일 다운로드 등 대용량 데이터가 이동하게 되며 이러한 서비스를 위해서는 비대칭적 트래픽 특성을 갖는 데이터를 효율적으로 처리할 수 있는 시스템이 요구된다. 3GPP의 TDD(Time Division Duplex) 방식은 CDMA를 기반으로 한 방식에 TDMA 개념을 도입하여 타임

슬롯의 적절한 할당을 통해 이러한 비대칭적 트래픽을 효율적으로 처리할 수 있는 장점을 가지고 있다.

무선이동 채널환경에서 멀티미디어 서비스를 제공하기 위해서는 채널환경 및 서비스 종류에 따라 다양한 방식의 에러보상기법이 요구된다. 에러보상 기법으로 채널 부호화 방식과 재전송 방식이 있는데, 채널 부호화 방식은 시간지연이 적고 송/수신단 사이에 별도로 주고 받는 정보가 필요 없다는 장점이 있지만 양호한 채널환경 하에서 시스템 효율이 감소하는 단점이 있다. 반면 재전송 방식은 시간지연이 생기게 되며 열악한 채널환경 하에서는 오히려 시스템 효율이 낮아지는 단점이 발생한다. 이러한 문제점을 해결하기 위해서 제안된 것이 Hybrid ARQ (Automatic Repeat reQuest) 방식으로, H-ARQ 방식에서는 채널 부호화와 재전송의 두 방식을 결합

하여 보다 효율적인 에러보상을 할 수 있다[1],[2].

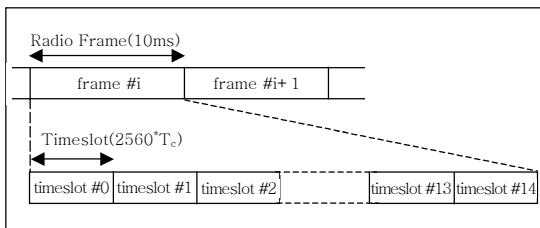
본 고에서는 멀티미디어 서비스를 위한 고속 패킷 전송 방식으로 3GPP TDD 방식을 고려하였다. TDD 시스템에서 효율적인 패킷 전송을 위하여 터보 부호를 사용한 다양한 H-ARQ 기법에 관하여 고찰하였으며, 이동통신 채널환경에서 각각 그 성능을 분석하였다.

II. CDMA/TDD 시스템

TDD 방식의 기본적인 물리 채널의 구조를 (그림 1)에 나타내었다. 10ms의 길이를 가지는 하나의 무선 프레임은 15개의 타임슬롯으로 구분되며 하나의 타임슬롯은 2560칩으로 이루어져 있다.

TDD 방식의 프레임은 15개의 타임슬롯으로 구성되어 있으며 업링크와 다운링크의 트래픽 특성에 따라 다양한 프레임의 구성할 수 있다. 특히 인터넷 서비스나 VOD 서비스와 같은 비대칭적인 트래픽 특성을 갖는 데이터를 전송할 때 주파수를 효율적으로 사용할 수 있게 된다.

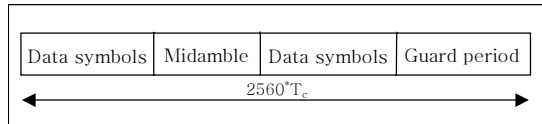
물리 채널은 할당된 무선 프레임 내의 타임슬롯을 통해 전송되는 버스트로 이루어져 있으며, 버스트의 주기는 하나의 타임슬롯이다. 몇 개의 버스트가 동일 타임슬롯에 동시에 전송될 수 있는데 이때, 동일 타임슬롯에 전송되는 버스트들은 서로 다른 OVVSF(Orthogonal Variable Spreading Factor) 채널 코드를 사용하여 구별하게 된다. 이와 같이 TDD 방식은 기존의 FDD 방식에서는 얻을 수 없던 채널 할당의 유연성을 가지고 있으며, CDMA와 TDMA 방식의 장점을 모두 취할 수 있는 구조를 가



(그림 1) TDD 방식의 물리 채널 구조

지고 있다[3]-[6].

전용 물리 채널에서는 세 종류의 버스트 타입이 정의되어 있으며, 버스트 구조는 (그림 2)와 같이 두 개의 데이터 심볼 필드, 미드엠블, 그리고 guard period로 구성되어 진다.



(그림 2) TDD 시스템의 버스트 구조

하나의 버스트에 전송되는 데이터 심볼의 수는 버스트 타입과 SF(Spreading Factor)에 따라 결정된다. <표 1>에 버스트 타입과 SF에 따른 버스트의 데이터 심볼 수를 나타내었다. 각각의 버스트는 서로 다른 동작 환경에 적합하게 정의되어 있다. 업링크 SF는 16, 8, 4, 2, 1이 사용되며 다운링크는 16과 1이 사용된다.

<표 1> 버스트 형태에 따른 데이터 심볼 수

SF	Type 1	Type 2	Type 3
1	1952	2208	1856
2	976	1104	928
4	488	552	464
8	244	276	232
16	122	138	116

미드엠블은 FDD 방식의 파일럿 심볼과 같이 채널 추정을 위해 사용되며, 동일 셀 내 동일 타임슬롯의 다수 사용자들은 하나의 기본 미드엠블 코드를 cyclic shift 시켜서 사용한다. 트래픽 버스트에서 미드엠블은 데이터 필드와 달리 확산이나 채널화 과정을 거치지 않는다. 기본 미드엠블 코드의 길이는 456칩(버스트 타입 1, 3) 또는 192칩(버스트 타입 2)이며 실수축과 허수축을 교번하는 복소수 열이다.

버스트의 미드엠블은 기본 미드엠블 코드를 512 칩 또는 256칩으로 주기적으로 확장하여 만든 주기적 기본 미드엠블 시퀀스를 사용하게 된다. 모두 128개의 기본 미드엠블 코드가 있으며 다른 셀의

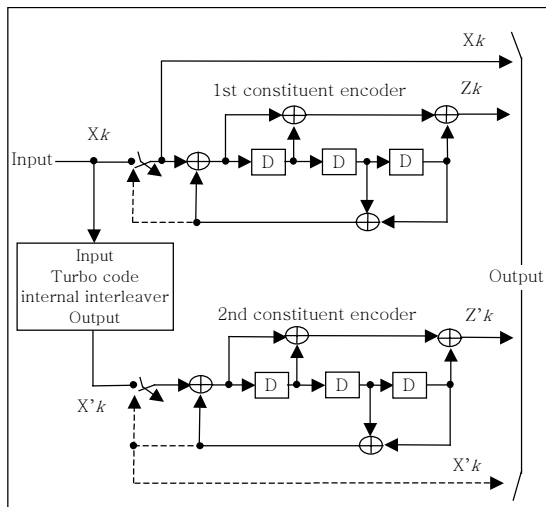
사용자는 다른 기본 미드엠블 코드를 사용하게 되고 동일 셀 내의 동일 타임슬롯의 사용자들은 주기적 기본 미드엠블 시퀀스를 각각 57칩 또는 64칩 만큼 시간지연 시켜 사용하게 된다. 따라서 업링크에서 하나의 타임슬롯에 할당될 수 있는 최대 사용자의 수는 각각 16 또는 6이 된다.

TDD 방식 시스템에서 채널 부호화 방식으로 R=1/2 또는 R=1/3의 길쌈부호와 R=1/3의 터보 부호가 사용된다.

길쌈 부호는 구속장 9로써 생성다항식은 R=1/2 부호의 경우 561(octal), 753(octal)이며, R=1/3 부호의 경우 557(octal), 663(octal), 711(octal)이다. 부호화 과정이 끝난 후 트렐리스를 제로 상태로 만들어 주기 위하여 8비트의 0 값이 추가된다.

터보 부호는 구속장 4로써 두 개의 8상태 구성 부호기와 내부 인터리버로 구성되어 있다. (그림 3)에 터보 부호기의 구조를 도시하였다.

각 구성부호기는 피드백 생성다항식 13(octal)과 피드 포워드 생성다항식 15(octal)로 이루어져 있으며, 부호기의 출력은 $x_1, z_1, z'_1, x_2, z_2, z'_2, \dots, x_K, z_K, z'_K$ 이다. 부호화 과정이 끝난 후 2개의 구성부호기의 트렐리스 상태를 제로 상태로 만들어 주기 위하여 (그림 3)의 점선부분 동작을 수행하며, 이때 부호기의 출력은 $x_{K+1}, z_{K+1}, x_{K+2}, z_{K+2}, x_{K+3}, z_{K+3}, x'_{K+1}, z'_{K+1}, x'_{K+2}, z'_{K+2}, x'_{K+3}, z'_{K+3}$ 과 같다.



(그림 3) 터보 부호기의 구조

부호화된 신호를 대역 확산하고 변조하기 위해서 이진 데이터 비트열을 복소 심볼로 매핑하여 준다.

01은 1, 00은 j, 10은 -1, 11은 -j로 매핑하고 매핑된 심볼열들을 대역 확산 시켜준다.

매핑된 복소 심볼들은 실수 값을 가지는 채널 코드에 의해 대역 확산된다. 채널 코드는 SF에 따라서 1, 2, 4, 8, 16의 길이를 가지며 이를 OVFSF 코드라고 부른다. OVFSF 코드는 같은 타임슬롯에 여러 가지 SF를 가진 채널들에 있어서 직교성을 유지하며 다중화 시켜준다.

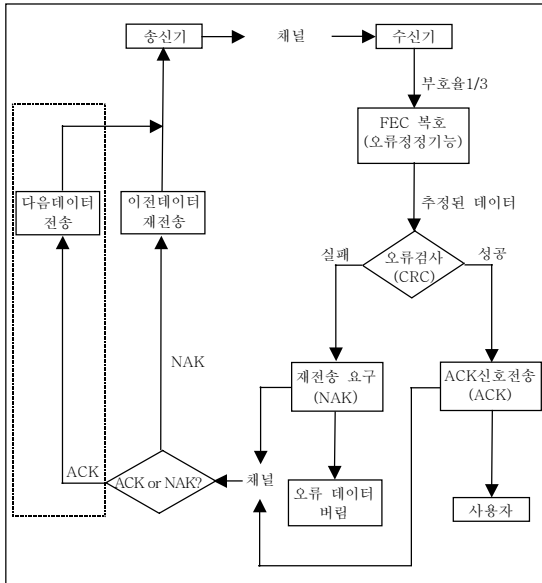
채널 코드로 대역 확산된 신호는 셀마다 고유한 스크램블링 코드에 의해 스크램블링 된다. 각 스크램블링 코드는 복소수의 값을 가지며 실수축과 허수축을 교번하는 주기 16칩을 가지는 복소수 열이다.

스크램블링 코드는 셀마다 고유하며 모두 128개가 있다. 모든 스크램블링 코드들은 다시 32개의 코드 그룹으로 분류되며 정해진 코드 그룹에 따라서 동기 채널의 t_{offset} 이 정해지게 된다.

III. H-ARQ 방식

1. H-ARQ Type I

(그림 4)에 H-ARQ Type I 방식의 블록 다이어그램을 나타내었다. 무선 이동 채널을 통과하여 수신된 신호는 복조기를 거쳐 가우시안 잡음이나 페이딩의 영향을 가진 값의 형태로 복호기로 전달된다. 복호기는 오류 정정 과정을 실행하고 추정된 데이터를 출력하여 주며, 추정된 데이터에 오류 존재 여부를 판단하여 오류가 포함되어 있지 않을 경우 사용자에게 전달한다. 만일 오류가 포함되어 있다고 판단된 경우에는 ARQ에 따라 재전송 과정이 시작되며, 이전에 수신된 데이터는 모두 버린다. 재전송을 통해 새로운 데이터가 수신되면 위의 과정을 반복하게 된다. H-ARQ Type I 방식은 채널 부호화된 simple-ARQ 방식이라고 할 수 있다. H-ARQ Type I의 변



(그림 4) H-ARQ Type I 블록 다이어그램

형된 방식으로 오류가 포함되었다고 판정된 데이터를 버리지 않고 저장하였다가 재전송된 데이터와 결합하는 기법이 있는데 이를 Chase combining 기법이라 한다.

H-ARQ Type I 방식에서 FEC에 사용되는 부호는 오류정정 기능과 동시에 오류검출 기능을 가져야 한다. 이는 ARQ를 사용하는 시스템에 있어서 필수적이며 오류정정을 위한 채널 부호 이외에 패리티 체크 비트(parity check bit)나 CRC(Cyclic Redundancy Check)와 같은 오류 검출 부호가 필요하다.

오류정정 부호를 사용하게 되면 각 전송과 재 전송에 있어서 오버헤드 부담을 증가시키게 되며, 그 결과로 채널환경이 양호한 상황에서는 오류정정 부호가 없는 simple-ARQ 방식을 사용할 때보다 전송 효율이 떨어진다. 그러나 채널환경이 열악해져 가면서 H-ARQ Type I 방식은 채널 부호가 가지는 오류정정 능력에 의해 simple-ARQ 방식보다 전송 효율이 높아지게 된다.

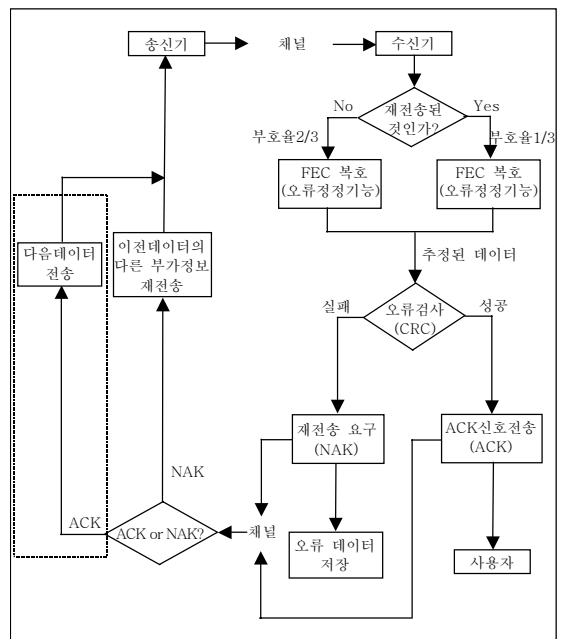
2. H-ARQ Type II

H-ARQ Type I 방식의 단점을 보완하고 데이터

전송 시 채널 부호에 의한 오버헤드 부담을 줄이기 위해 여러 정정기능을 위한 부가정보(redundancy)를 수신기의 요구에 따라 적절히 가변하여 전송하는 ARQ 방식을 H-ARQ Type II 방식이라고 한다[7].

(그림 5)에 H-ARQ Type II 방식의 블록 다이어그램을 나타내었다. 전송될 데이터의 부호화는 부호율 1/3로 실행한 후, 전송에 의해 기본 부호율 2/3인 부호를 사용하여 데이터를 전송한다. 페딩 채널과 같이 시간에 따라서 변하는 채널 환경에서 채널 상황이 양호한 경우 2/3 부호와 같은 적은 양의 부가 정보만으로도 데이터 오류정정이 가능하게 되며 시스템적인 측면에서는 전송되는 부가 정보의 양이 적으므로 시스템 효율이 높아진다.

채널 상황이 열악하여 이미 수신된 부호율 2/3의 부호화된 비트만으로 오류정정이 되지 못한다면, ARQ를 통해 재전송이 이루어지게 된다. 재전송 시 H-ARQ Type I 방식이 동일한 부호화 비트들을 전송했던 것과는 달리, 이미 수신된 비트를 제외한 나머지 부호화된 비트를 전송하여 기존의 수신된 정보와 결합하여 부호율 1/3의 형태로 오류정정을 실행하게 된다.



(그림 5) H-ARQ Type II 블록 다이어그램

H-ARQ Type II 방식은 전술한 방법과 같이 재전송 시 항상 증가된 부가정보를 전송하게 되므로 Full IR 방식이라고 불리우기도 한다. 이 방식은 채널 환경에 따라서 적절하게 부가되는 정보의 양을 조절할 수 있으므로 보다 효율적인 재전송을 구현할 수 있게 된다.

그러나 송신기가 현재 송신하지 않은 나머지 부호화된 비트들을 수신기가 요구할 때를 대비하여 항상 버퍼에 저장하고 있어야 하고 수신기도 재전송 요구 후에 오류 발생 패킷을 결합하기 위하여 앞서 수신했던 부호화된 비트들을 버퍼에 저장하고 있어야 한다는 단점이 있다. 또한 IR을 구현하기 위하여는 H-ARQ Type II 방식에 적합한 채널 부호기 및 천공 패턴을 찾아내야 한다.

3. H-ARQ Type III

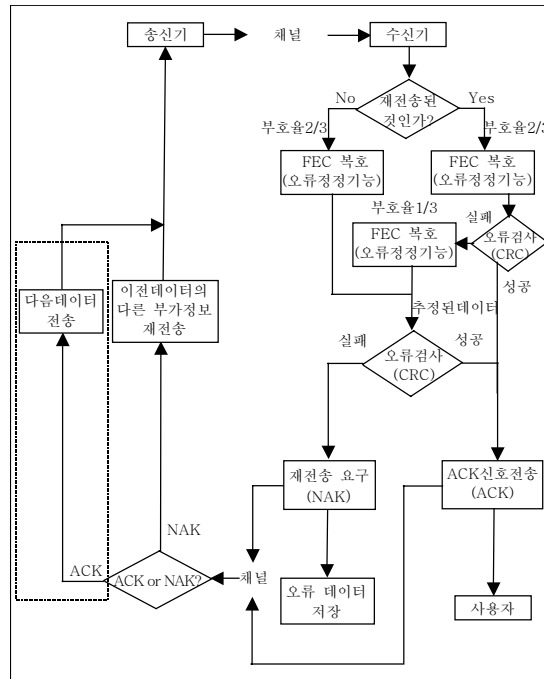
H-ARQ Type II 방식에서 재전송 시 전송되는 부가적인 정보는 일반적으로 그 자체만으로는 복호가 가능하지 않다. 따라서 복호기는 재전송된 부가 정보 이외에 처음 전송되었던 패킷에 의존해야만 한다. 만일 첫번째 전송된 패킷이 손실되었거나 간섭 등에 의해 심하게 손상되었을 경우 H-ARQ Type II 방식은 커다란 성능 열화를 가져온다. 따라서 이러한 경우 전송된 첫번째 패킷에 의존하지 않고 재전송 되어진 정보만으로 복호가 가능하다면 좀더 향상된 성능을 획득할 수 있게 된다. 이러한 특성을 가지는 H-ARQ 방식을 H-ARQ Type III라고 하며, HSDPA에서는 Partial IR 이라고도 한다. (그림 6)에 H-ARQ Type III 방식의 블록 다이어그램을 나타내었다.

H-ARQ Type III 방식의 구현을 위하여 재전송되는 부가정보가 self-decodable한 특성을 가지는 부호들을 도입해야 한다. 예를 들면, 길쌈 부호를 기반으로 한 시스템에서 CPC(Complementary Punctured Convolutional) 코드라는 개념의 부호를 도입하여 H-ARQ Type III 방식을 구현할 수 있다[8]. CPC 코드는 재전송 시 전송되는 각 부호들이 동일

한 에러 정정 능력을 가지도록 천공 되었으며 천공된 각각의 부호를 결합하게 되면 천공 이전의 원 부호 특성을 나타낼 수 있도록 설계되었다.

H-ARQ Type III는 H-ARQ Type II와 마찬가지로 처음 송신 시의 부호율을 채널 상황에 맞게 설정한 뒤 전송하게 되며 복호 후에 오류가 검출 되었다면 오류가 검출된 패킷을 버리지 않고 저장한 뒤 재전송을 요구한다. 재전송된 신호를 수신하게 되면 H-ARQ Type II와 달리 재전송된 부가정보만으로 복호를 수행하므로 처음 수신된 오류 패킷에 의존하지 않고 복호할 수 있게 된다. 만일 재전송된 패킷만으로 복호를 수행하였으나 오류가 발생하였다면 처음 전송된 패킷과 결합하여 낮은 부호율을 가지는 부호를 생성한 후 복호 과정을 다시 한번 거치게 되므로 에러 정정 능력이 향상되게 된다.

따라서 H-ARQ Type III 방식은 다중 사용자에 의한 순간적인 간섭의 증가, 급격한 페이딩 상황의 변화에 따른 패킷의 심한 손상이나 손실에 따른 성능 열화에 좀더 효과적인 H-ARQ 방식이라 할 수 있다.



(그림 6) H-ARQ Type-III 블록 다이어그램

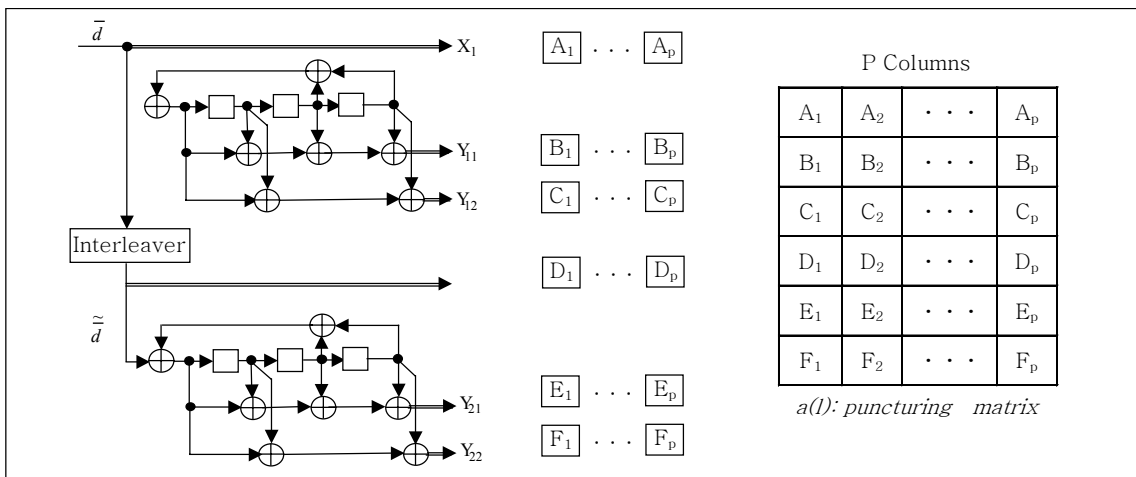
4. RCPT 코드

H-ARQ를 구현하기 위해서는 각 H-ARQ 방식에 적합한 채널 부호의 집합이 필요하다. TDD 방식의 시스템에서는 채널 부호로 터보 부호가 사용되고 있는데 다양한 방식의 H-ARQ를 구현하기 위해 RCPT 부호를 고려한다. RCPT 부호는 RCPC(Rate Compatible Punctured Convolutional) 코드와 같은 개념의 부가 정보를 제공하여 주는 부호의 종류이며, 터보 부호를 사용하는 시스템에서 다양한 부호율의 터보 부호를 제공함으로써 H-ARQ를 가능하게 하여 준다[9],[10].

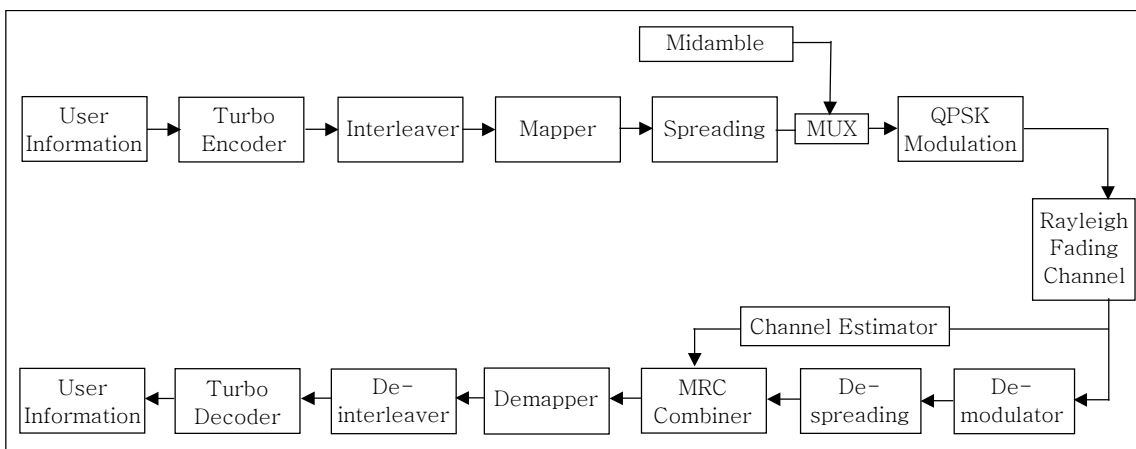
(그림 7)에 하나의 인터리버를 사용한 모부호율 1/6인 RCPT 부호를 나타내었다.

구현된 RCPT 코드는 P(puncturing period), R(code rate)에 따라서 puncturing matrix $a(l)$ 로 나타난다. 구현된 RCPT 코드를 이용하여 Full IR, Partial IR, Chase combining 등의 다양한 H-ARQ를 구현할 수 있다.

본 고에서는 하나의 패킷에 대한 최대 전송횟수를 세 번으로 설정하여 첫번째, 두번째, 세번째의 전송패턴을 나타내주는 $a(0)$, $a(1)$, $a(2)$ 을 기본 코드율과 H-ARQ 방식에 따라서 정의하고 시스템의 스루풋을 분석하였다.



(그림 7) 하나의 인터리버를 사용한 RCPT 코드



(그림 8) TDD 시스템 모의실험용 블록 다이어그램

IV. 모의실험

(그림 8)에 TDD 시스템 모의 실험을 위한 블록 다이어그램을 나타내었으며, <표 2>에 모의실험에 사용된 파라미터들을 나타내었다.

<표 2> TDD 시스템 모의실험에 사용된 파라미터

Chip Rate	3.84Mcps
Interleaver	16 by 16 block interleaving
Spreading	Spreading factor = 16
Burst Type	Type I
Channel Model	Rayleigh fading, 1, 2path Jakes Power Spectrum Mobile Speed: 10km/hr
Channel Estimator	Ideal Channel Estimator Non-ideal Channel Estimator

본 모의실험에서는 3.84Mcps의 광대역 TDD 시스템을 고려하여 시뮬레이터를 구성하였다. 버스트 타입 I에서 guard period는 없다고 가정하고 GP 필드를 데이터 필드에 포함하여 2560칩으로 이루어진 하나의 타임슬롯 단위로 데이터 프레임 구성하였다.

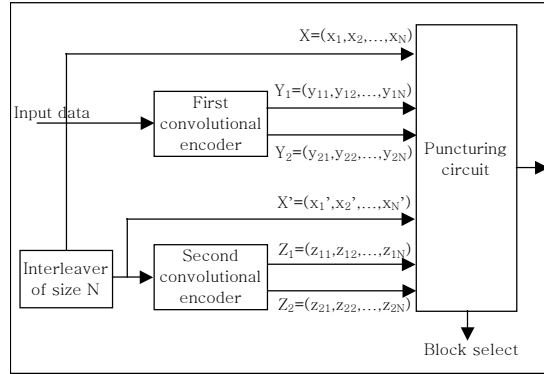
고려된 채널은 레일리 페이딩 채널이며 다중 경로로 페이딩 환경을 고려하여 주파수 선택적 페이딩 채널을 구현하였다. 다중 경로는 1path와 2path를 고려하였으며 각기 1칩 만큼의 시간지연을 갖는 것으로 가정하였고, 2path인 경우 각 경로의 에너지 크기는 각각 0.5, 0.5를 가지는 것으로 가정하였다.

UE의 이동성을 고려하여 도플러 주파수 천이가 고려된 페이딩 프로세스를 발생시키기 위하여 Jakes Power Spectrum을 가지는 페이딩 프로세스가 UE의 속도 10km/hr에 의해 발생하였다.

채널 추정기는 non-ideal 채널 추정기를 사용하였는데, 이는 단순 평균 출력을 제공하는 FIR 필터로 이루어져 있고 512칩의 미드앰블을 참조하여 슬롯 단위의 채널 추정치를 제공하여 준다.

이러한 TDD 시스템 시뮬레이터를 이용하여 여러 가지 H-ARQ 방식, 즉 Chase combining, Full IR, Partial IR 방식의 시스템 효율을 분석한다.

H-ARQ를 위한 터보 부호기를 (그림 9)에 도시



(그림 9) H-ARQ에 사용된 터보 부호기 구조

하였다. 3GPP TDD 방식에 정의되어 있는 터보 부호기는 최소 1/3의 부호율을 가지고 있으나, 다양한 부호율을 가지는 H-ARQ를 구현하기 위하여 최소 1/5의 부호율을 가지는 터보 부호를 사용한다.

<표 3>에 H-ARQ 모의 실험을 위한 파라미터를 나타내었다. 최대 전송횟수를 3회로 하고 각 H-ARQ는 stop-and-wait 방식을 사용한다.

<표 3> H-ARQ 모의실험 파라미터

부호율	1/2, 2/3, 3/4
구속장	K: 4(3 memory, 8 state)
생성다항식	(13, 15, 17)
최대 전송횟수	3회
ARQ 방식	stop-and-wait

<표 4>에 Full IR과 Partial IR 방식의 재전송 횟수에 따른 최종 부호율을 나타내었다. Partial IR의 경우 반복된 패킷의 부분은 부호율에 고려하지 않고 재전송 시 추가된 새로운 부가정보만을 부호율에 고려하였다.

<표 4> 전송횟수와 부호율과의 관계

	H-ARQ 타입	첫번째 전송	두번째 전송	세번째 전송
1/2 base code	Full IR	1/2	1/4	1/6
	Partial IR	1/2	1/3	1/4
3/4 base code	Full IR	3/4	3/8	1/4
	Partial IR	3/4	3/5	1/2

<표 5>와 <표 6>은 부호율과 H-ARQ 방식에 따른 전송 패킷을 나타낸다.

<표 5> R=1/2 base code의 천공 패턴

(a) Full IR의 천공 패턴

	첫번째 전송 a(0)		두번째 전송 a(1)		세번째 전송 a(2)	
	data bit	tail bit	data bit	tail bit	data bit	tail bit
X	11	111 000	00	111 000	11	111 000
Y1	10	111 000	01	000 000	00	111 000
Y2	00	000 000	10	111 000	01	000 000
X'	00	000 111	00	000 111	00	000 111
Z1	01	000 111	00	000 000	10	000 111
Z2	00	000 000	11	000 111	00	000 000

(b) Partial IR 천공 패턴

	첫번째 전송 a(0)		두번째 전송 a(1)		세번째 전송 a(2)	
	data bit	tail bit	data bit	tail bit	data bit	tail bit
X	11	111 000	11	111 000	11	111 000
Y1	10	111 000	01	000 000	10	111 000
Y2	00	000 000	00	111 000	00	000 000
X'	00	000 111	00	000 111	00	000 111
Z1	01	000 111	10	000 000	01	000 111
Z2	00	000 000	00	000 111	00	000 000

(c) Chase combining의 천공 패턴

	data bit	tail bit
X	11	111 000
Y1	10	111 000
Y2	00	000 000
X'	00	000 111
Z1	01	000 111
Z2	00	000 000

<표 6> R=3/4 base code의 천공 패턴

(a) Full IR의 천공 패턴

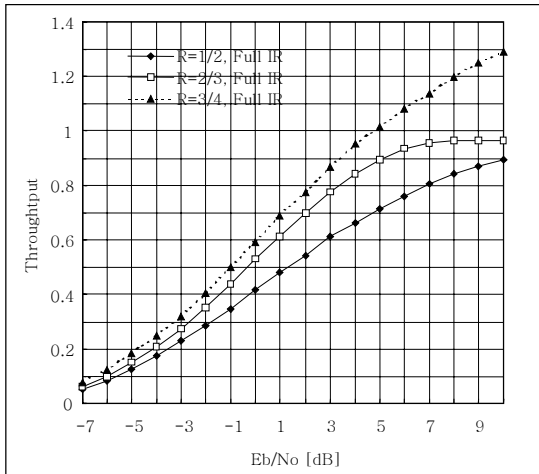
	첫번째 전송 a(0)		두번째 전송 a(1)		세번째 전송 a(2)	
	data bit	tail bit	data bit	tail bit	data bit	tail bit
X	111 111	111 000	000 000	000 000	000 000	111 000
Y1	001 000	001 000	100 010	110 000	001 001	000 000
Y2	000 000	000 000	010 100	101 000	001 001	010 000
X'	000 000	000 000	000 000	000 000	000 000	000 111
Z1	000 001	000 111	001 001	000 110	100 010	0000 00
Z2	000 000	000 001	001 001	000 101	010 100	000 010

(b) Partial IR의 천공 패턴

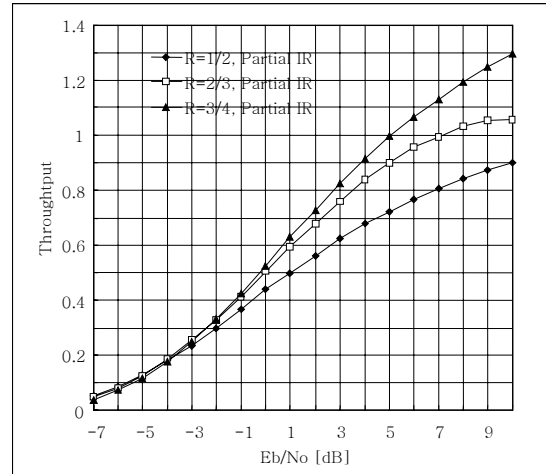
	첫번째 전송 a(0)		두번째 전송 a(1)		세번째 전송 a(2)	
	data bit	tail bit	data bit	tail bit	data bit	tail bit
X	111 111	111 000	111 111	111 000	111 111	111 000
Y1	001 000	001 000	000 000	110 000	001 000	000 000
Y2	000 000	000 000	001 000	101 000	000 000	010 000
X'	000 000	000 000	000 000	000 000	000 000	000 111
Z1	000 001	000 111	000 000	000 110	000 001	000 000
Z2	000 000	000 001	000 001	000 101	000 000	000 010

(c) Chase combining의 천공 패턴

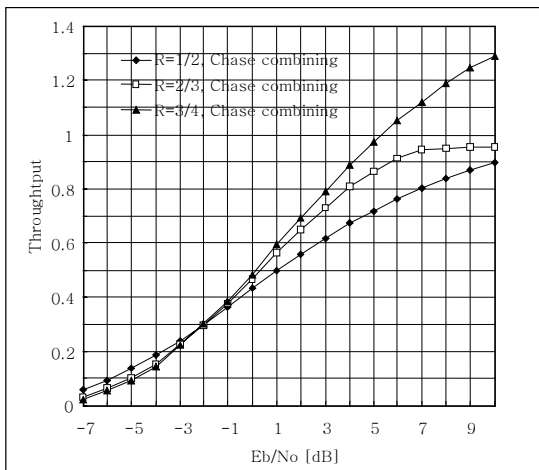
	data bit	tail bit
X	111 111	111 000
Y1	001 000	001 000
Y2	000 000	000 000
X'	000 000	000 000
Z1	000 001	000 111
Z2	000 000	000 001



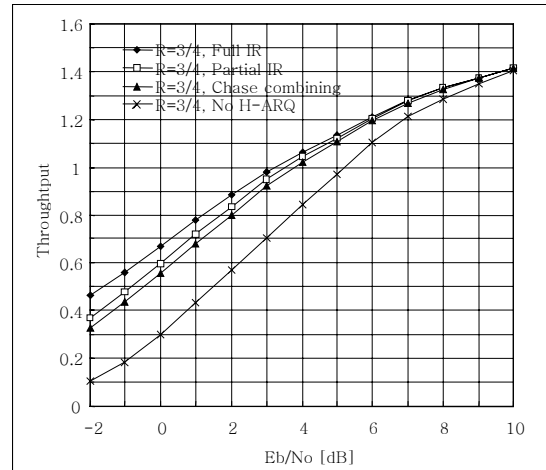
(그림 10) TDD 시스템에서 Full IR의 주파수 효율 (v=10km/hr, 1path)



(그림 12) TDD 시스템에서 Partial IR의 주파수 효율 (v=10km/hr, 1path)



(그림 11) TDD 시스템에서 Chase combining의 주파수 효율(v=10km/hr, 1path)



(그림 13) R=3/4 기반의 H-ARQ 방식 주파수 효율 (v=10km/hr, 2path)

TDD 시스템의 업링크에서 사용자는 하나의 프레임에서 하나의 타임슬롯을 사용하여 패킷을 전송하는 방식을 가정하였다. 부호화된 프레임은 한 타임슬롯에 해당하는 길이를 가지며, 오류 발생 시 재전송은 다음 프레임의 타임슬롯으로 전송됨을 가정하여 시스템 효율을 분석하였다.

(그림 10)에서 (그림 12)는 UE의 속도가 10km/hr일 경우 각 H-ARQ 방식의 부호율에 따른 TDD 시스템의 주파수 효율을 나타내었다.

높은 부호율 기반의 방식이 낮은 부호율 기반의

방식에 비해 이득은 Full IR 방식이 Partial IR, Chase combining 방식에 비하여 더욱 크다. (그림 11)에서 Chase combining의 경우 낮은 Eb/No에서 R=1/2 부호의 주파수 효율이 R=2/3, R=3/4의 경우보다 약간 높는데 이는 재전송 시 결합된 부호의 에러 정정 능력이 초기 전송된 부호의 에러 정정 능력에 의해 크게 좌우되는 Chase combining의 특성이 잘 나타나고 있기 때문이다.

(그림 13)에서는 다중 경로를 고려한 경우의 각 H-ARQ 방식에 따른 주파수 효율을 알아보기 위한

그림을 도시하였다.

다중 경로를 고려한 경우에도 각 H-ARQ 사이의 성능 차이에 관한 경향은 하나의 경로만을 고려한 경우와 유사하게 나타난다. 각 H-ARQ별 성능차이는 높은 부호율 기반의 방식에서 크게 나타나며 Full IR > Partial IR > Chase combining 순서의 성능을 보여준다.

V. 결론

본 고에서는 멀티미디어 서비스를 위한 고속 패킷 전송 방식으로 3GPP TDD 방식을 고려하여 터보 부호를 사용한 H-ARQ 기법에 관하여 그 성능을 분석하였다.

TDD 시스템에서 여러 가지 H-ARQ 방식을 적용해 시스템 효율을 분석한 결과, 시스템의 주파수 효율은 Full IR > Partial IR > Chase combining > No H-ARQ의 순서대로 나타난다. 부호율에 따른 성능을 살펴보면 전반적으로 높은 부호율을 기반으로 하는 H-ARQ의 성능이 우수하게 나타나게 된다. Full IR의 경우 R=3/4를 기반으로 하는 방식이 R=2/3, R=1/2을 기반으로 하는 방식에 비해 커다란 이득을 보이며, 이는 R=3/4을 기반으로 하는 Full IR의 경우 재전송되어 나타나는 부호의 에러 정정 능력의 향상이 R=2/3, R=1/2 방식에 비해 크기 때문이다. Chase combining과 Partial IR의 경우도 R=3/4 기반의 방식이 다른 방식에 비해 높은 주파수 효율을 보이지만 그 차이는 크지 않다.

Full IR 방식은 전 Eb/No 영역에서 Chase combining, Partial IR 방식보다 우수한 성능을 보여주지만 R=1/2의 낮은 부호율 기반 방식에서는 이득이 크지 않으며 수신측의 버퍼링 부담이 가장 크다는 단점이 있다.

부호율의 변화에 따른 시스템 효율에서 전반적으로 R=3/4 기반의 H-ARQ 방식이 R=2/3, R=1/2 기반의 H-ARQ 방식에 비하여 우수한 주파수 효율을

보여 준다. 그러나 평균 재전송횟수는 R=3/4 기반의 H-ARQ 방식이 낮은 부호율 기반의 H-ARQ 방식에 비해 크다는 단점이 있다.

고려된 H-ARQ 방식 중에서 R=3/4을 기반으로 한 Full IR 방식이 전반적인 성능이 가장 우수한 결과를 보였지만 평균 재전송횟수가 늘어나고 수신측 메모리가 많이 필요한 단점이 있다.

참고 문헌

- [1] S. Lin and D. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, Prentice Hall, 1983.
- [2] J. Hagenauer, "Rate-Compatible Punctured Convolutional Codes(RCPC Codes) and Their Applications," *IEEE Trans. Commun.*, Vol. 36, Apr. 1988, pp. 389 - 400.
- [3] 3GPP TS 25.221: "Physical Channels and Mapping of Transport Channels onto Physical Channels (TDD)."
- [4] 3GPP TS 25.222: "Multiplexing and Channel Coding(TDD)."
- [5] 3GPP TS 25.223: "Spreading and Modulation (TDD)."
- [6] 3GPP TS 25.224: "Physical Layer Procedures (TDD)."
- [7] H. Lou *et al.*, "Performance of Punctured Channel Codes with ARQ for Multimedia Transmission in Rayleigh Fading Channels," VTC, 1996, pp. 282 - 286.
- [8] TIA/EIA/IS-2000. 2(Ballot Version).
- [9] Krishna R. Narayanan, and Gordon L. Stuber, "A Novel ARQ Technique Using the Turbo Coding Principle," *IEEE Commun. Letters*, Vol. 1, No. 2, Mar. 1997.
- [10] Douglas N. Rowitch and Laurence B. Milstein, "On the Performance of Hybrid FEC/ARQ Systems Using Rate Compatible Punctured Turbo (RCPT)Codes," *IEEE Trans. Commun.*, Vol. 48, June 2000, pp. 948 - 958.