

# 단축 부호를 이용한 적응적 복합 ARQ 기법

正會員 김 영 섭\*, 박 세 응\*\*

## An Adaptive Hybrid ARQ Scheme Using Shortened Codes

Youngsup Kim\*, Saewoong Bahk\*\*, *Regular Members*

### 요 약

본 논문에서는 기존의 ARQ와 복합 ARQ (hybrid ARQ) 기법을 위한 하위 블록 재전송기법을 제안한다. 제안하는 하위블록 재전송 기법은 채널의 상태가 양호할 때는 기존의 ARQ와 복합 ARQ 기법과 똑같이 동작한다. 채널의 상태가 나빠짐에 따라, 전송되는 데이터 블록은 효과적인 전송을 위해 작은 하위 블록으로 나뉘어 진다. 각각의 하위 블록은 오류 정정을 위해 단축 부호를 이용하여 부호화 되고 부호의 길이는 채널의 BER에 따라 최적의 크기로 선택된다. 수신된 블록은 하위 블록 단위로 오류를 검사한다. 제안하는 하위 블록 재전송 기법은 오류 발생 시 하위 블록만을 재전송함으로써, 기존의 ARQ 기법보다 향상된 통과율을 얻을 수 있다. 하위 블록기법을 사용하여 ATM 셀을 전송하는 예를 고찰하고 시뮬레이션 한다.

### ABSTRACT

In this paper we propose a sub-block retransmission scheme for ARQ and hybrid ARQ schemes. When the channel is quiet the sub-block retransmission scheme behaves like a conventional ARQ or hybrid ARQ scheme. As the channel is becoming noisy, the data block is divided into small sub-blocks for transmission. Each sub-block is encoded for error control by an appropriate shortened code and the code length of the shortened code is being adaptive to the corresponding channel BER. The received block is checked for errors sub-block by sub-block. The proposed sub-block retransmission scheme provides improved throughput over conventional schemes by retransmitting only the naked sub-blocks in the occurrence of errors. An example of transferring ATM cells is considered for simulation.

\*한국이동통신 PCS추진본부 시스템개발팀

\*\*서울대학교 전기공학부  
SoEE, Seoul National Univ.  
論文番號:96004-0103  
接受日字:1996年 1月 3日

## I. 서론

기본적으로 오류제어를 위한 기법에는 두 가지가 있는데, 자동 반복 요구(ARQ)와 순 방향 오류 정정(FEC)기법이 있다. ARQ 기법은 단순하면서도 높은 신뢰도를 보장한다. 하지만, 통과율이 일정하지 않고, 채널의 상태가 나빠지면 통과율은 급격히 감소한다. FEC 기법은 일정한 통과율을 보장하지만, 채널의 상태가 나빠짐에 따라 기법의 안정성이 떨어진다. 복합 ARQ 기법은 ARQ와 FEC를 결합한 방식이다. 복합 ARQ 기법은 크게 Type-I과 Type-II로 나눌 수 있다. Type-I 방식이 Type-II 방식보다 간단하다. Type-I 기법에서는 데이터 블록을 오류의 검출과 수정을 위해 부호화 한다. 수신단에서는 먼저 수신된 데이터에 대해서 오류 정정을 한다. 수신된 데이터가 오류 정정 가능 범위의 오류인 경우, 오류 정정을 수행한 뒤 수신단에 전달하며, 그렇지 못한 경우 재전송을 요구한다. Type-II 기법은 변하는 채널의 상황에 동적으로 적용한다. 이 방식에서 재전송 요구를 받은 송신단은 처음 송신시의 부호를 재전송 하지 않고, 중복도에 해당되는 부호를 재전송한다. 수신단에서는 이전에 수신된 데이터에 다시 재전송된 부호를 덧붙여서 오류 정정을 하게 된다[1][2]. Type-II 기법은 넓은 범위의 채널 비트 오류율을 갖는 환경에서 잘 동작한다.

이 논문에서는 기존의 ARQ와 복합 ARQ를 위한 하위 블록 재전송 기법을 제안한다. 채널의 잡음이 없을 때, 하위 블록 재전송 ARQ (SRS-A: Subblock Retransmission Scheme for ARQ)는 기존의 ARQ처럼 동작한다. 채널의 잡음이 증가함에 따라, 데이터 블록은 전송을 위해 작은 하위 블록으로 나누어진 다. 각각의 하위 블록은 단축부호를 사용하여 부호화 되고 부호의 길이는 채널의 BER에 따라 적응적으로 변한다. 수신단에서는 수신된 블록에 대하여 하위 블록 단위로 오류를 검사한다. 오류가 발생했을 때, 하위 블록만을 재전송함으로써 향상된 통과율을 얻을 수 있다.

II장에서 본 연구와 관련된 오류 제어 기법을 살펴본다. 순(pure) ARQ과 복합 ARQ에 대해 살펴보고, 하위 블록 재전송 기법을 사용함으로써 기존의 기법들에 비해 향상된 통과율을 얻을 수 있음을 보인다. III장에서는 하위 블록 재전송 기법에 있어서 단일 부

호 복합 ARQ 기법이 가장 적절한 방법임을 보인다. ATM 셀을 전송하는 모의 실험을 보이며 IV장에서 결론을 맺는다.

## II. 하위 블록 재전송 기법과 다른 기법들과의 비교

주어진 채널의 오류환경에 맞춰 부호의 길이를 적응적으로 변화시킬 때, 기존의 ARQ 기법과 복합 ARQ 기법이 향상된 통과율을 보인다는 것은 잘 알려진 사실이다[3, 4, 8]. [3]의 연구에서 Wu는 채널의 환경에 따라 다른 오류 정정능력을 갖는 다중 단축부호를 사용하는 복합 ARQ 기법 (MSC-HA)을 제안했다. 이 기법은 본 논문에서 제안하는 하위 블록 재전송 기법과 밀접하게 관련되어 있으므로, 잠시 살펴보기로 하자. 이 기법은  $l$  오류정정 가능  $(n+m, n)$  순회부호인  $C_1$ 으로부터 얻어진 단축부호를 사용한다. 원 부호를 부호화하고 복호화하는 같은 하드웨어를 사용하여 단축부호의 부호화, 복호화를 할 수 있다. 송신기는  $k=n-m$  비트의 블록을 전송하기 위해 데이터를 모은다. 송신단은  $S_0, S_1, S_2, \dots, S_\nu$ 로 표시되는 상태에서 동작을 한다. 채널의 비트 오류율  $p$ 에 따라서 동작 상태가 결정된다. 채널의 잡음이 없으면 수신기는 상태  $S_0$ 에서 동작한다.  $S_0$ 에서  $k$  비트 데이터 블록은  $n$  비트로 부호화 되는데,  $C_1$ 으로부터 얻어지는  $(n, n-m)$  단축 부호  $C_0$ 를 사용하여 부호화 한다.  $C_0$ 는 기존의 ARQ에서와 같이 오류의 검출을 위해서만 사용한다. 채널의 잡음이 심해짐에 따라 동작 상태를 변화시킨다.  $S_1$  상태에서  $k$  비트 데이터는  $C_0$ 를 사용하여  $n$  비트로 부호화 되고, 이  $n$  비트의 블록은 오류의 정정을 위해서  $C_1$  부호를 사용하여  $n+m$  비트의 단어로 부호화 한다. 이때  $S_1$ 에서의 동작은 기존의 복합 ARQ 기법과 동일하다.  $j=2, 3, 4, \dots, \nu$  일 때,  $S_j$ 는  $S_1$ 에서처럼  $C_0$ 에 기반하여  $k$ 비트의 데이터를  $n$ 비트로 부호화 한다. 이제 이  $n$ 비트의 데이터는 길이  $k_{j1}, \dots, k_{j\ell_j}$ 를 갖는  $\ell_j$ 개의 부분으로 나눈다. 나누어진 각 부분은 오류의 정정을 위해  $C_1$ 으로부터 얻어지는 단축 순회 부호를 이용하여 블록의 크기가  $k_{ji} + m (1 \leq i \leq \ell_j)$ 이 되는 하위 블록으로 부호화 한다. 따라서 전송 블록의 크기는  $n + \ell_j m$  비트가 된다. 수신단에서는 수신된 블록에 포함된 각각의 하위 블록에 대해서 오류 정정을 시도한다. 오류정정된 각각의

하위 블록을 모아서  $n$  비트의 데이터를 구성한다. 이러한  $n$  비트의 데이터에 대하여  $C_0$ 를 사용하여 오류를 검출한다. 만일 오류가 검출되면 수신단은 송신단에 재전송을 요구한다.

$C_0$ 를 이용해서 얻은 1003 비트의 부호어에 대한 예를 살펴보자. 1003 비트의 부호어를 상태  $S_2, S_3, S_4, S_5$ 에서 각각 2, 4, 8, 16 부분으로 나눈다. 그림 1은 각각의 상태에 대해서 버퍼의 크기가 무한대인 경우 선택적 반복 ARQ를 사용할 경우에 대한 통과율을 나타내고 있다.  $C_1$ 으로는 두 비트 오류정정 (1023, 1003) BCH를 사용하고,  $C_1$ 으로 부터 얻어지는 (1003, 983) 단축부호  $C_0$ 를 사용한다. 다중 단축 부호 복합 ARQ (MSC-HA)와 하위 블록 재전송 복합 ARQ (SRS-HA)를 비교하기 위해, 그림 1의 그래프에서 통과율은 이상적인 통과율의 곡선이 되는 경우로 가정하는데, 이는 통과율의 곡선의 포락선을 따라 가는 경우이다.

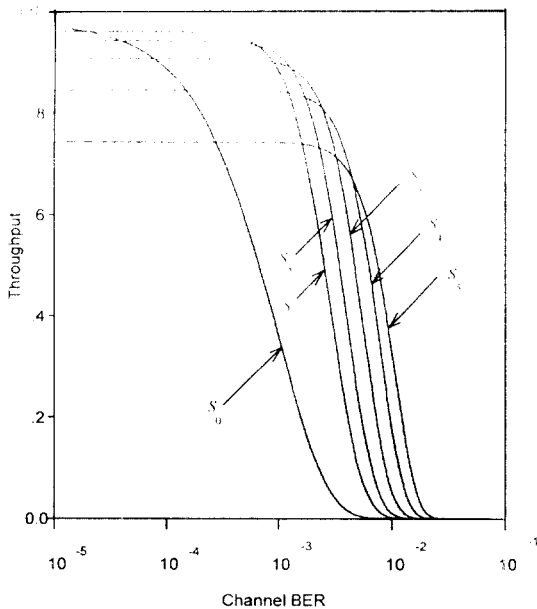


그림 1. 상태  $S_0 \sim S_5$ 에서 MSC-HA기법의 통과율(throughput).

Fig 1. Throughput versus BER for MSC-HA with states  $S_0 \sim S_5$ .

이제 MSC-HA와 SRS-HA를 비교한다. MSC-HA는 하위 블록단위로 오류 정정을 한다. 상태  $S_j$ 에서

오류의 검출은 오류정정이 끝난 복호화된  $\ell_j$ 개의 하위 블록을 모아서 만든 전체의 블록에 대하여 행한다. 만일 복호화된 하위 블록에 대해서 오류가 검출이 되면 전체 블록에 대한 재전송을 요구한다. 논문에서 제안하는 기법은 오류의 검출은 MSC-HA와 같은 방법을 사용하지만, 전체 블록에 대하여 재전송을 요구하지 않고 하위 블록단위의 재전송을 요구하는 차이점을 갖는다. MSC-HA와 SRS-HA를 비교하는 과정은 다음과 같다. SRS-HA는 MSC-HA와 같은  $C_1$ 을 사용하고, 이  $C_1$ 으로 부터 단축 부호가 만들어진다. 또한 두 방법 모두 같은 개수의 상태를 사용한다. SRS-HA는 상태  $S_0, S_1$ 에서 MSC-HA와 똑같이 동작한다.  $j=2, 3, 4, \dots, \nu$ 인 상태  $S_j$ 에서  $k$  비트의 데이터 블록은 각각의 길이  $k'_1, k'_2, \dots, k'_\ell$ 을 갖는  $\ell_j$ 개의 부분으로 나누어진다. 각각의 부분은 오류의 정정을 위해,  $C_1$ 으로 부터 얻어지는 단축부호를 사용하여  $k'_{j\ell} + m$  ( $1 \leq i \leq \ell_j$ )의 길이를 갖는 단어로 부호화 한다.  $C_1$ 으로 부터 얻어진 단축부호를 사용하여, 위에서 얻어진  $(k'_{j\ell} + m)$  비트의 단어를 오류 정정을 위하여  $k'_{j\ell} + 2m$ 의 길이를 갖는 단어로 부호화 한다.  $k'_{j\ell} + 2m$  비트를 갖는  $\ell_j$ 개의 하위 블록으로 이루어진 전체 블록은 전송 데이터 크기가  $k + 2 \ell_j m$ 가 된다. 수신단에서 부호화된 전체 블록을 받은후, 오류의 정정을 위해서 각각의 하위 블록을 복호화한다. 복호화된 하위 블록에 대해서 오류의 검출을 행한다. 만일 오류가 검출이 되면, 전체 블록에 대한 재전송을 요구하는 것이 아니라, 하위 블록에 대해서만 재전송을 요구한다. 수신단에서는 수신된 하위 블록에 대한 ack 정보를 담은 피드백 메시지를 송신단에 보낸다. 피드백 메시지는  $\ell_j$ 비트맵으로 구성된다.  $i$ 번째 비트는  $i$ 번째 하위 블록에 대한 ack/nak 정보를 담고 있다. 이 피드백 메시지를 이용하여, 송신단에서는 오류가 있는 하위 블록을 재전송한다.

그림 2에서 무한 크기의 버퍼와 선택적 재전송요구 방법을 사용하는 경우, 두가지 복합 ARQ기법에 대한 통과율을 보여준다. 하위 블록에 대한 피드백 메시지로 인해, MSC-HA기법보다 오버헤드가 증가하지만, 오류가 있는 하위 블록만을 재전송함으로써, MSC-HA보다 더 나은 성능을 보여줌을 알 수 있다. MSC-HA기법의 통과율 곡선은 그림 1에서 곡선들의 포락선이다. SRS-HA기법에서 983 비트의 데이터 블록은 각각

의 상태  $S_2, S_3, S_4, S_5$ 에 따라서, 2, 4, 8, 16 부분으로 나누어 진다. 복합 ARQ기법의 통과율 분석에 있어서 일반적으로 프로토콜 헤더의 영향은 무시함으로, 본 논문에서도 헤더의 영향은 무시한다. SRS-HA 기법은 MSC-HA기법보다 작은 크기의 단축부호를 사용함으로, 오류가 검출되지 않을 확률이 낮아진다. 따라서, MSC-HA보다 더 신뢰도가 높아진다[5].

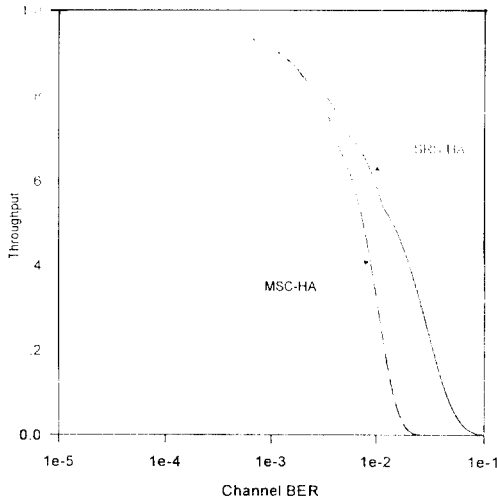


그림 2. SRS-HA와 MSC-HA의 수율의 비교.  
Fig 2. Throughput comparison of SRS-HA with MSC-HA.

이제 순 ARQ기법에 대해서 알아보자(순 방향 오류 제어가 없는 경우). 이 경우에도 하위 블록에 대한 오류 검출의 기법이 사용될 수 있는데, 전체 블록을 나누어 생긴 하위 블록 크기는 채널의 상태에 따라 정해지며, 최소 기본 크기의 정수배가 되도록 잡는다 [4]. [4]의 연구에서, Martins는 추정된 채널의 비트 오류율에 대해 최적인 블록의 크기를 사용하였다. 일단 블록의 크기가 정해지면, Martins이 사용한 기법은 기존의 ARQ기법처럼 동작한다. 헤더와 데이터에 대해서 오류를 검사하면, 오류가 검출된다. 잡음이 심한 환경에서는 헤더에 대해서, 데이터에 대한 오류 제어 방법과는 다른 오류 제어 방법을 사용한다. 이런 방법은 ATM이나 Internet Protocol(IP)과 같은 많은 프로토콜에서 사용되는데, 헤더에 오류가 발생하면, 패킷이 잘못된 목적지에 전달되기 때문이다. 헤더에 대해서 데이터와는 다른 방법으로 오류를 제어하게 되

는 잡음이 심한 환경에서, 데이터 영역에도 오류 제어 기법을 사용할 때, SRS-A가 Martins의 방법보다 성능이 우수한 것은 분명하다. SRS-A는 하위 블록으로 이루어진 전체의 블록에 대해서 하나의 헤더가 있는 반면, Martins의 방법은 각각의 하위 블록마다 헤더가 있기 때문이다.

그림 3에서 알 수 있듯이, SRS-A기법이 넓은 범위의 채널 비트 오류율에서 Martins의 방법보다 우수하게 동작한다. 여기서 CRC와 헤더의 길이는 각각 16, 32 비트다. SRS-A의 데이터 블록의 크기는 1024 비트로, 채널의 잡음이 없는 경우에 Martins의 ARQ방법에서 사용되는 가장 길이가 긴 데이터의 크기다. Martins의 ARQ에서, 1024, 512, 256, 128, 64 비트의 5가지 길이의 데이터를 사용한다. 나중에 유도 될 이상적인 통과율의 곡선은 이 5가지 데이터 크기에 대한 통과율 곡선들의 포락선이 된다. 1024 비트의 데이터 블록은 같은 크기를 갖는 2, 4, 8, 16개의 블록으로 나뉘고, 각각은 16 비트의 CRC로 부호화 된다. 헤더는 오류가 없고, 피드백 채널에는 잡음이 없다고 가정한다. 두 가지 기법의 안정성은 같다.

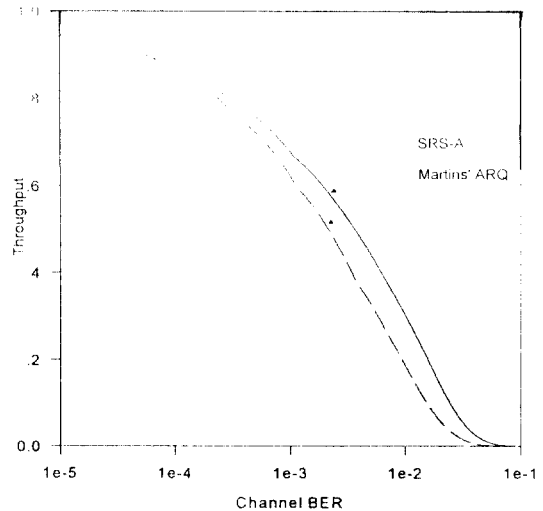


그림 3. SRS-A와 Martins ARQ의 수율의 비교.  
Fig 3. Throughput comparison of SRS-A with the Martins, ARQ.

### III. 단일 부호 복합 ARQ기법에서 하위 블록 재전송 기법

이번 장에서는 하위 블록을 재전송하는 단일 부호 복합 ARQ기법(SRS-SCHA)에 대해서 살펴보는데, 이는 단일 부호가 오류 제어에 효과적이며 하위 블록 재전송 기법의 응용에 가장 적합하기 때문이다[2]. 송신단은  $S_1, S_2, \dots, S_\nu$  ( $S_0$  가 없음)로 표시되는 상태중 하나의 상태에서 동작한다. 동작 상태는 이전의 경우처럼 채널의 비트오류율에 따라 결정된다.  $t$  비트의 오류정정이 가능하고 최소거리  $d_{\min}$ 을 갖는  $(n_0 + m, n_0)$  순회부호로 부터 얻어지는  $(n_0 + m - \ell, n_0 - \ell)$  단축 부호를  $S_s(\ell)$ 이라 하자. 송신기는 전송을 위해  $K$  비트를 갖는 블록을 만든다.  $d_1, d_2, \dots, d_\nu$  ( $d_1 < d_2 < \dots < d_\nu$ )를  $K$ 의 약수라 하자.  $K$  비트를 갖는 데이터 블록은 상태  $S_j$ 에서  $d_j$  ( $1 \leq j \leq \nu$ )개의 부분으로 나뉜다. 각 부분은  $(n_0 + m - \ell, n_0 - \ell)$  단축 부호  $C_s(\ell)$ 를 사용하여  $n_0 + m - \ell$ 비트로 부호화 시킨다.  $\ell$  값은 다음의 식으로 얻어진다.

$$n_0 - \ell = K/d_j \quad (1)$$

$d_j$ 개의 하위 블록으로 이루어진 부호화 된 전체 블록을 전송하는 제안하고자 하는 SRS-SCHA의 동작순서는 다음과 같다.

- 부호화된 블록이 수신되면, 각각의 하위 블록은  $t'(\leq t)$ 의 오류정정 가능 한정 거리 복호화기를 사용하여 복호화 한다.
- 수신기는 수신기의 상태를 나타내는 피드백 메시지를 수신단에 전달한다. 수신 기가 보내는 피드백 메시지는 전체 블록과 하위 블록에 대한 ack/nack를 포함한 다. 전체 블록에 대한 ack/nack는 수신단에서 전체 블록이 받아 들여졌는지를 표시한다. 이 부분은 기존의 ARQ기법에서 ack가 하는 역할을 한다. 하위 블록에 대한 ack/nack 부분은  $d_j$ 개의 비트로 이루어진 비트맵이다. 이 비트맵에서  $i$ 번째 비트는  $i$ 번째 하위 블록의 ack/nack를 표시한다. 복호화기에서 복호화에 실패하면, 이 하위 블록에 대해서 nack를 보낸다. 이 경우는 하위 블록의 오류가 검출은 되지만, 정정될 수 없는 상태를 나타낼 때이다.

- 수신된 전체 블록에 포함된 하위 블록중 어느 한 하위 블록이 nack를 받으면, 수신 단은 상태 메시지와 제대로 받은 하위 블록을 수신단의 버퍼에 저장한다.
- 전송된 전체 블록중에 ack를 받지 못한 하위 블록의 수를  $r(\leq d_j)$ 라고 하자. 송신단에서 수신단으로 부터 피드백 메시지를 받으면, 송신단은 ack를 받지 못한  $r$ 개의 하위 블록을 순서대로 재전송한다.
- 재전송한  $r$ 개의 하위 블록이 실패 없이 복호화되면,  $r$ 개의 하위 블록과 이전에 수신되어 수신단의 버퍼에 저장된  $(d_j - r)$ 개의 하위 블록은 상위 계층으로 넘겨지고 버퍼를 비우게 된다.
- 만약 재전송후에도 ack를 받지 못하는 하위 블록이 있는 경우, 위에서 행한 재전송 과정을 반복한다.

이제 제안한 기법의 성능을 분석하자. ATM 망에 기반한 무선망에서 ATM 셀을 전송하는 것을 고려한다. ATM 셀은 5 바이트의 헤더와 48 바이트의 데이터로 구성된다. ATM 헤더는 [7]에서 제안되었듯이 무선환경에 맞춰 4 바이트의 무선 ATM 헤더로 바뀌어 사용할 수 있다. 무선 망에서의 ATM 헤더는 전송되는 셀의 순서인 순서번호, SN를 포함한다. 하위 블록 재전송 기법을 48 바이트의 ATM 데이터 전송에 적용한다. 무선망에서의 ATM 헤더는 다른 오류 제어 기법을 사용하여 보호하고, 32 비트(4 바이트)의 헤더는  $h(\geq 32)$  비트로 부호화 하고, 수신단에서 올바르게 수신되었다고 가정한다.  $n = n_0 + m - \ell$ ,  $k = n_0 - \ell$ 인  $(n, k)$  단축 부호  $C_s(\ell)$ 을 생각하자.  $t'(\leq t)$  오류 정정 가능 한정거리 복호화기를 사용한다. 부록에서 유도되는 SRS-SCHA의 통과율은 다음 식으로 주어진다.

$$\eta = K \left[ nM + h + n \frac{MP_r}{1 - P_r} + h \sum_{L=1}^M \binom{M}{L} \sum_{r=1}^{M-L} \binom{M-L}{r} (-1)^r \frac{P_r^{L+r}}{1 - P_r^{L+r}} \right]^{-1} \quad (2)$$

여기서,  $M$ 은  $K$  비트의 블록이 나누어진 하위 블록의 수이며,  $P_r$ 은 하위 블록이 재전송될 확률이다.

피드백 채널은 잡음이 없다고 가정한다. 수신단에서 받아들여진 블록의 오류 확률  $P(E)$ 는 수신단에서

받아들여진 전체 블록에 대한 오류로 받아들여진 블록의 비로 주어진다. 이 식의 값은 다음과 같다.

$$P(E) = 1 - \left( \frac{P_c}{P_c + P_{ud}} \right)^M \quad (3)$$

$P_c$ 는 수신된 단어가 올바르게 복호화 될 확률이고,  $P_{ud}$ 는 복호화시 오류를 검출하지 못할 확률이다. (2), (3)의 식은 block code[2]의 분석에 주어져 있다.

$$P_r = 1 - (P_c + P_{ud}) \quad (4)$$

$$P_c = \sum_{i=0}^{l'} \binom{n}{i} p^i (1-p)^{n-i} \quad (5)$$

$$P_{ud} = \sum_{i=d_{min}}^n A_i P_i \quad (6)$$

여기에서  $A_i$ 는 무게  $i$ 를 갖는 부호어의 개수이며,  $P_i$ 는 수신된 단어가 송신된 부호어로부터 Hamming 거리가  $i$ 인 단어에 중심을 두고 있는 반지름이  $l'$ 인 Hamming구 안에 들어갈 확률이다. 이 값은 다음의 식으로부터 주어진다.

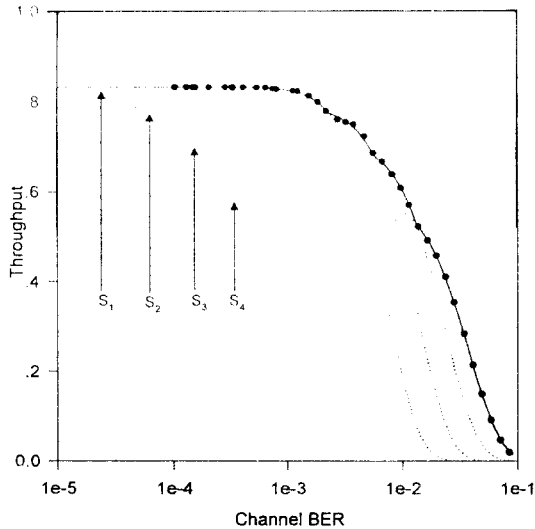


그림 4.  $S_1 \sim S_4$ 에서 SRS-SCHA의 수율 곡선.  
Fig 4. Throughput curves of SRS-SCHA with states  $S_1 \sim S_4$ .

$$P_i = \sum_{v=0}^{l'} \sum_{r=0}^v \binom{i}{v-r} \binom{n}{r} p^{i-v-2r} (1-p)^{n-i+v-2r} \quad (7)$$

4가지의 상태를 갖는 예를 들어보자. 부호화된 ATM 헤더는 50 비트라고 가정한다.  $S_1$ 상태에서, 48 바이트 ( $K=384$  비트)의 데이터 블록은 3 비트 오류정정 가능 (511, 484) BCH 부호  $C$ 로 부터 얻어지는 (411, 384) 단축부호  $C_s(100)$ 를 사용하여 부호화 한다. 이 데이터 블록은 각각 상태  $S_2, S_3, S_4$ 에서 2, 4, 8 부분으로 나누어진다. 각각의 하위 블록은 적절한 단축 부호를 사용하여 부호화 하는데,  $S_2$ 에서는 (219, 192) 단축 부호  $S_s(292)$ ,  $S_3$ 에서는 (123, 96) 단축 부호  $C_s(388)$ ,  $S_4$ 에서는 (75, 48) 단축 부호  $C_s(436)$ 을 사용한다. 수신 단에서는 2비트 오류 정정 가능 한정거리 복호화기를 사용한다. 그림 4에서 이 예에 대한 통과율을 보여주고 있다. 그림 5에서는 받아들여진 셀의 오류율을 보여주고 있다. 받아들여진 블록의 오류율은 ARQ의 신뢰도를 나타낸다.

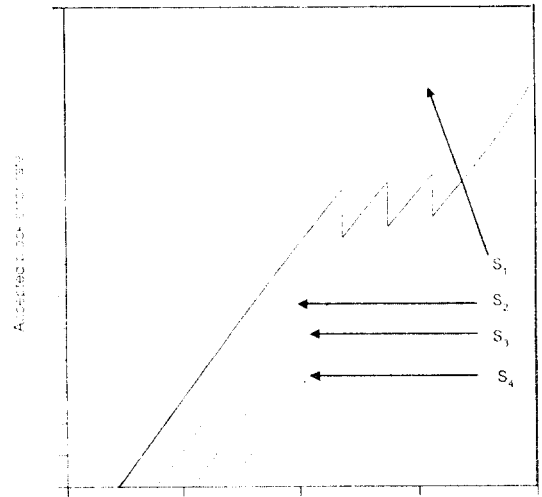


그림 5. 상태  $S_1 \sim S_4$ 에서 받아들여진 블록의 오류율.  
Fig 5. Accepted block error rate of SRS-SCHA with states  $S_1 \sim S_4$ .

#### IV. 결 론

본 논문에서는 순 ARQ와 복합 ARQ를 위한 하위 블록 재전송 기법을 제안했다. 제안된 하위 블록 재

전송 기법은 오류 발생시 하위 블록만을 재전송함으로써, 기존에 제안된 ARQ기법보다 좋은 성능을 보여 줄을 알 수 있었다. 예로써, 하위 블록 재전송 복합 ARQ (SRS-HA)기법을 ATM 셀에 적용해 보았다. 동작상태를 결정하는 방법을 제안하고 모의 실험을 하였다. 모의 실험 결과 제안한 기법을 사용한 경우 통과율은, 이상적인 통과율 고선을 매우 근접하게 따라 가는 것을 볼 수 있었다. 제안된 기법은 단축 부호를 사용하고 변하는 채널의 BER에 동적으로 적응함으로써, 통과율과 신뢰도를 높였다.

순 방향 오류정정(FEC)과 [8]에서 사용된 하드웨어를 사용할 때, 제안된 기법은 무선 ATM 링크에서 요구되는 다양한 오류 제어 능력을 갖는다. 또한 하위 블록 재전송 기법은 재전송 되는 횟수를 제한함으로써, 시간 제약이 있는 응용 프로그램의 데이터 전송에 사용될 수 있다. 하위 블록 재전송 기법은 기존의 stop and wait (SW)와 go back N (GBN)의 재전송 기법에도 적용시킬수 있다. 이 기법들과 함께 사용하는 경우의 성능의 향상 정도를 알아보기 위해서 더 많은 연구가 필요하겠다. 하위 블록 재전송 기법은 기존의 ARQ기법보다 긴 피드백 메시지를 사용하는데, 하위 블록에 대한 ack/nack를 담은 비트맵을 포함하기 때문이다. 피드백 메시지의 크기가 약간 증가하지만, 전체적인 성능 향상을 고려하면 제안된 기법은 충분히 효과적이라고 볼 수 있다.

**부 록**

식 (2)의 유도

순 ARQ와 복합 ARQ의 통과율은 다음 식으로 주어진다.

$$\eta = \frac{K}{E[T]} \tag{A.1}$$

K는 주어진 블록의 비트수고, T는 주어진 블록의 전송을 위해서 필요한 비트수를 표시하는 불규칙 변수다. E[·]는 주어진 값에 대한 평균값을 나타낸다. T는 다음 식으로 주어진다.

$$T = (Mn + h) + \sum_{i=1}^{\infty} T_i \tag{A.2}$$

$K_i$ 는  $i$ 번째 재전송시 전송되는 비트수를 나타내는 불규칙 변수를 나타낸다. 주어진 블록에 대하여 전송되는 비트수의 평균값은 다음 식으로 주어진다.

$$E[T] = (Mn + h) + \sum_{i=1}^{\infty} E[T_i] \tag{A.3}$$

이제  $i$ 번째 재전송시 전송되는 비트수의 평균값  $E[T_i]$ 의 값을 계산하자. 불규칙 변수  $T_i$ 는  $i$ 번째 재전송시  $M$ 개의 하위 블록중  $L$ 개의 하위 블록을 재전송하는 경우  $Ln + h$ 의 값을 갖는다.  $i$ 번째의 재전송시 재전송되는 하위 블록은  $i$ 번동안 계속 오류가 검출되는 경우다. 이 사건에 대한 확률을  $P_r^i$ 라고 하면,  $i$ 번째의 재전송에서  $M$ 개의 하위 블록 중  $L$ 개의 하위 블록이 재전송될 확률은 다음 식으로 주어진다.

$$\Pr\{T_i = Ln + h\} = \binom{M}{L} P_r^{iL} (1 - P_r^i)^{M-L} \tag{A.4}$$

$E[T_i]$ 는 다음 식으로 주어진다.

$$\begin{aligned} E[T_i] &= (Mn + h) + \sum_{i=1}^{\infty} \sum_{L=1}^M (Ln + h) \binom{M}{N} P_r^{iL} (1 - P_r^i)^{M-L} \\ &= (Mn + h) + n \sum_{i=1}^{\infty} \sum_{L=1}^M L \binom{M}{N} P_r^{iL} (1 - P_r^i)^{M-L} \\ &\quad + h \sum_{L=1}^M L \binom{M}{L} \sum_{i=1}^{\infty} P_r^{iL} (1 - P_r^i)^{M-L}. \end{aligned} \tag{A.5}$$

이항 불규칙 변수의 평균값에 대한 식 (A.6)과 등식 (A.7)을 (A.5)식에 적용하면, (A.8)식을 얻을 수 있다.

$$\sum_{k=1}^n k \binom{n}{k} p^k (1-p)^{n-k} = \sum_{k=1}^n k \binom{n}{k} p^k (1-p)^{n-k} = np \tag{A.6}$$

$$(1+x)^n = \sum_{k=0}^n \binom{n}{k} x^k \tag{A.7}$$

$$\begin{aligned} T &= (Mn + h) + n \sum_{i=1}^{\infty} M P_r^i \\ &\quad + h \sum_{L=1}^M \binom{M}{L} \sum_{i=1}^{\infty} P_r^{iL} \sum_{r=0}^{M-L} \binom{M-L}{r} (-P_r^i)^r \\ &= Mn + h + nM \sum_{i=1}^{\infty} P_r^i \\ &\quad + h \sum_{L=1}^M \binom{M}{L} \sum_{r=0}^{M-L} \binom{M-L}{r} (-1)^r \sum_{i=1}^{\infty} P_r^{(L+r)i} \end{aligned} \tag{A.8}$$

$$= Mn + h + \frac{nMP_r}{1-P_r} + h \sum_{L=1}^M \binom{M}{L} \sum_{r=0}^{M-L} \binom{M-L}{r} (-1)^r \frac{p_r^{L+r}}{1-p_r^{L+r}}$$

참 고 문 헌

1. S. Lin and D. J. Costello, Jr., *Error Control Coding : Fundamentals and Applications*, New Jersey:Prentice-Hall, 1983.
2. S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, New Jersey:Prentice-Hall, 1995.
3. K. Wu, S. Lin, and M. Miller, A Hybrid ARQ Scheme Using Multiple Shortened Cyclic Codes, in *Proc. IEEE GLOBECOM82*, pp. 712-716, 1982.
4. J. A. C. Martins and J. D. C. Alves, ARQ Protocols with Adaptive Block Size Perform Better Over a Wide Range of Bit Error Rates, *IEEE Trans. Commun.*, vol. 38, No. 6, pp. 737-739, June 1990.
5. T. Fujimura, T. Kasami, A. Kitai, and S. Lin, On the Undetected Error Probability for shortened Hamming Codes, *IEEE Trans. Commun.*, vol. 33, no. 6, pp. 570-574, June 1985.
6. D. Bertsekas and R. Gallager, *Data Networks*. New Jersey:Prentice-Hall, 1992, 2nd ed.
7. D. Raychaudhuri and N. D. Wilson, ATM-Based Transport Architecture for Multiservices Wireless Personal Communication Networks, *IEEE J. Select. Areas Commun.*, vol. 12, no. 8, pp. 1401-1414, Oct. 1994.
8. D. Moore and M. Rice, Variable Rate Error Control for Wireless ATM Networks, in *Proc. IEEE ICC95*, pp. 988-992, 1995.



김 영 섭(Young Sup Kim) 정회원  
 1994년: 서울대학교(전자공학과 학사)  
 1996년: 서울대학교(전자공학과 석사)  
 1996년~현재: KMT PCS 추진본부 시스템 개발팀



박 세 웅(Sae Woong Bahk) 정회원  
 1962년 1월 11일생  
 1984년: 서울대학교 전기공학과(학사)  
 1986년: 서울대학교 전기공학과(석사)  
 1991년: University of Pennsylvania(박사)  
 1991년~1994년 2월: AT&T 벨 연구소 연구원  
 1994년 3월~현재: 서울대학교 전기공학부 조교수  
 ※주관심분야: 고속 통신 프로토콜, 통신망 성능분석