

# 인터넷 폰 서비스의 손실패킷 복구를 위한 인터리빙 FEC 메카니즘

정회원 황진수\*, 최승권\*\*, 안상규\*\*\*, 조용환\*\*

## Interleaving FEC Mechanism for Packet Loss Recovery of the Internet Phone Services

Jin-Soo Hwang\*, Seung-Kwon Choi\*\*, Sang-Kyu Ahn\*\*\*, Young-Hwan Cho\*\* *Regular Members*

### 요 약

본 논문에서는 매체독립 FEC와 주로 무선통신에서 사용되는 인터리빙 기법의 장점을 접목한 인터리빙 FEC를 제안하였다. 패킷 손실 복구에 인터리빙 기법을 사용하는 이유는 손실의 영향을 분산시킴으로써 패킷 복구의 가능성을 높일 수 있다. 그러나 모든 데이터가 인터리빙 된다면 이는 심각한 패킷 지연을 야기한다. 따라서 인터리빙 FEC에서는 지연을 최소화하기 위해 양여정보에 대해서만 인터리빙을 수행한다. 시뮬레이션 결과 인터리빙 FEC는 현재 인터넷 폰에서 널리 사용하고 있는 FEC에 비해 높은 복구율을 나타냈다. 특히 패킷 손실율이 높거나 연속적인 패킷손실이 발생하는 경우 더욱 유용하게 사용될 수 있다.

### ABSTRACT

This paper proposes an interleaving FEC technique which has combined the advantages of media-specific FEC scheme and interleaving technique, which is being mainly used for wireless communications. By use of interleaving technique, the possibility to recover lost packets can be much increased due to the interleaving characteristics to disperse the effect of packet losses. Nevertheless, in case whole data packets were interleaved, the delay would become a major problem as per the increase of interleaved segments. In this context, the proposed interleaving FEC technique has adopted the interleaving of repair data only, while data packets are being sent in ordinary sequence, to minimize the delay time during transmission. The results of simulation show that the proposed interleaving FEC method has marked higher recovery rate than conventional FEC approach, which is widely being used for internet phone services. In particular, the proposed method will be more effective in case of burst packet losses and higher packet loss environment.

### I. 서 론

인터넷 폰은 패킷음성기술의 발전으로 새롭게 부각된 서비스로 기존 공중전화망에 비해 매우 저렴한 가격으로 이용할 수 있으며 응용 프로그램에 따라 매우 다양한 서비스를 제공할 수 있는 고부가가치의 통신매체로 발전하고 있다. 인터넷 폰은 이미 15개 이상의 외국업체들이 제품을 발표하고 있으며,

현재 국내에서는 데이콤, 한국통신, 아이넷 등의 통신사업자가 서비스를 제공하고 있다.

인터넷 폰에서 패킷 손실 발생시 패킷 복구를 위하여 사용하는 기법은 기본적으로 ARQ와 FEC가 사용된다. ARQ는 재전송을 기반으로 하기 때문에 네트워크 지연이 발생하여 인터넷 전화와 같은 실시간 데이터 전송 서비스에 적용하기 어려우며, 반면에 FEC는 실시간 서비스에 적용시킬 수는 있으나

\* 에릭슨 코리아

\*\* 충북대학교 컴퓨터공학과(yhcho@cubucc.chungbuk.ac.kr)

\*\*\* 한국에너지기술연구소

논문번호 : 00118-0412, 접수일자 : 2000년 4월 12일

비교적 커다란 전송률을 요구한다<sup>[1]</sup>.

현재 이러한 복구기법만으로는 인터넷 폰을 위한 충분한 품질을 보장할 수 없으며 이에 대한 새로운 패킷 손실 복구기법이 필요하다. 손실 패킷의 복구율을 높이기 위한 방법으로는 잉여정보를 부가한 후 인터리빙을 수행하는 등의 여러 가지 방법이 제안되었으나 모두 지연이 심각하게 커지는 등의 단점을 가지고 있어 실시간 서비스인 인터넷 폰에서 사용하기에는 무리가 따른다.

이에 따라 본 논문에서는 기존 전화 서비스와 인터넷 폰 서비스의 비교시 가장 중요한 보완점으로 지적되고 있는 서비스 품질 향상을 위해 손실 패킷 복구기법을 검토한 후, 개선된 방안의 패킷 복구 기법을 제시하고, 제안한 인터리빙 FEC 기법이 손실 패킷 복구율을 증가시킬 수 있음을 입증하고자 한다.

## II. 손실패킷 복구기법 제안

### 2.1 손실 패킷 복구 기법 개요

실시간 데이터를 인터넷을 이용하여 전송할 경우 발생하는 특성으로는 종단간 지연, 지연의 변화인 지터, 패킷 손실, 비순차 패킷 등이 있다. 종단간 지연은 대부분 수신측에서 버퍼링을 행하고, 재생시간을 지연시킴으로써 상당수 해결할 수 있지만 패킷 손실을 복구하기 위해서는 추가적인 복구 알고리즘이 필요하다. 또한 패킷의 크기나 전송률에 따라서도 패킷 손실 특성이 크게 달라지므로 이에 대한 고려도 충분히 이루어져야 한다.

이와 같이 인터넷 상에 음성 데이터를 실시간으로 전송하였을 때 만약 트래픽이 심하다면 목적지의 수신자가 수신정보를 해독할 수 없을 정도의 패킷 손실이 나타날 수 있다. 이러한 경우를 위하여 여러 제어 메커니즘이 필요하고 여러 가지 에러 복원 메커니즘이 발표되고 있는데 대표적인 것은 ARQ (Automatic Repeat reQuest)와 FEC(Forward Error Correction)이며 이 외에도 ECC(Error Correction Code)를 이용한 방법과 인터리빙(Interleaving)을 이용한 방법이 최근 발표되었다.

여기서는 오디오 스트림의 패킷 손실을 복구하기 위한 송신자 기반의 복구방법을 논한다<sup>[1]</sup>.

이들 기술은 크게 두 가지로 구분된다. 능동적인 재전송과 수동적인 채널코딩이다. 수동적인 채널코딩은 다시 전통적인 FEC를 포함하는 채널 코딩 기술과 인터리빙 기반 기법들로 나뉘어 진다. FEC 데이터는 전형적으로 exclusive-or(XOR) 연산에 기반

한 매체 독립적인 형태이거나, 음성 신호의 특성에 기반한 매체 종속적인 형태일 수도 있다. 이들의 분류는 그림 1에 잘 나타나 있다.

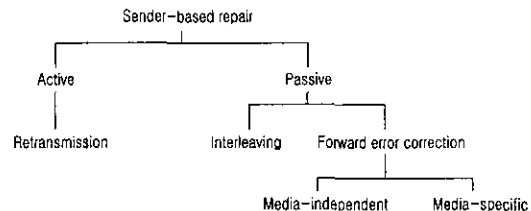


그림 1. 송신자 기반 복구 기법의 분류

향후 논의를 위해 패킷과 데이터의 유닛 구분하기로 한다. 데이터 유닛은 오디오 톨에 내부적으로 저장되는 오디오 데이터의 단위로 패킷은 하나 또는 그 이상의 유닛으로 이루어지며 네트워크 상에 전송되기 위해 캡슐화된다.

#### 2.1.1 Forward Error Correction

FEC는 손실된 패킷을 복원하기 위하여 송신측에서 원래의 정보에다 잉여정보를 추가하여 전송하고 수신단에서 이 잉여정보를 이용하여 손실패킷을 복구하는 개루프 메커니즘이다. FEC는 최소한의 지연만으로 에러 복원과 신뢰성을 확보할 수 있는 메커니즘이기 때문에 실시간 전송이 반드시 요구되는 응용분야에 있어서는 매우 중요한 기술이다. 하지만 FEC의 패킷 손실에 대한 복구능력은 네트워크 상에서의 패킷 손실 특성 즉 손실 패턴에 크게 의존하게 되므로 네트워크 상태나 트래픽 상황에 맞게 손실복구 알고리즘을 개발할 필요가 있다.

이에 따라 전송중에 데이터의 손실을 복구하기 위해 여러 FEC 기술들이 개발되어 왔다. 이러한 기법은 손실된 패킷을 복구할 수 있도록 부가적인 데이터를 스트림에 더하는 방식이다. 스트림에 부가되는 복구 데이터는 두 가지로 구분된다. 즉 스트림 내용에 관계없이 독립적인 데이터와 스트림 내용에 관한 정보를 담고 있는 데이터로 구분된다.

##### 1) 매체 독립 FEC

손실 복구를 위한 부가적인 패킷을 생성하기 위해 블록 코드나 대수 코드를 사용하는 매체 독립 FEC에 대한 관심이 높아지고 있다. 각 코드는 k-데이터 패킷의 코드워드를 가지며 네트워크에서 n 패킷의 전송을 위한 n-k 부가 check 패킷을 생성한다. 여러가지 블록 코딩 기법이 있지만 여기서는 현재 RTP의 payload로 제안된 parity 코딩과 Reed-

Solomon 코딩의 두 가지만 논한다<sup>[2]</sup>. 이러한 블록 코딩 기법은 원래 정송된 bit들의 스트림내에서 에러를 검지하고 정정하기 위해 설계되었으므로 check bit들은 데이터 bit들의 스트림으로부터 생성된다. 패킷 스트림에서 모든 패킷의 손실을 고려해 보면 패킷의 블록에 적합한 bit에 블록코딩 기법을 적용한다. 따라서 check 패킷의 i번째 bit는 관련된 데이터 패킷의 i번째 bit로부터 생성된다.

패리티 코딩에서는 exclusive-or(XOR)연산이 적합한 패리티 패킷을 생성하는데 적용된다. 이것의 예는 Rosenberg에 의해 구현되었다<sup>[3]</sup>. 이 방법에서 하나의 패리티 패킷은 각 n-1 데이터 패킷의 뒤에 전송된다. 각 n 패킷중에서 단지 하나의 손실이 발생하면 이는 복구될 수 있다.

Reed-Solomon 코드는 에러 복구 특성이 우수하며, 특히 연속된 손실에 대한 복구력이 뛰어나다<sup>[4,5]</sup>. 그러나 패킷 손실이 없는 디코딩은 인코딩과 같은 계산 비용을 가지지만 손실이 발생할 경우 심각한 비용을 발생시킨다.

FEC 기법은 손실된 패킷이 송신지에서 목적지까지 보내진 패킷 스트림 전체에 퍼졌을 때, 즉 패킷 손실이 연속적이지 않고 가능한 간헐적으로 발생할 때 더욱 효과를 얻을 수 있다. 이러한 FEC 기법은 기본적으로 n개의 패킷 내에 하나의 패킷 손실이 생긴 것을 복원할 수 있으며 상대적으로 간단한 방법이지만 1/n 만큼의 소스의 전송률을 증가시켜야 하고 손실된 패킷을 재생하기 전에 n 패킷을 받아야 하기 때문에 그만큼의 지연을 증가시켜야 하며, 디코더의 구현이 어려운 단점이 있다. 장점으로는 매체에 독립적이라는 것이다. 즉, FEC의 동작은 패킷의 내용에 의존하지 않으며 복구는 손실 패킷을 완전히 대체한다. 게다가 에러 정정 패킷의 계산이 작으며 구현이 쉽다.

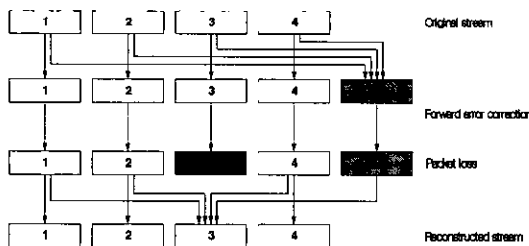


그림 2. 패리티 FEC를 이용한 복구

2) 매체 종속 FEC

패킷 손실을 보호하는 간단한 방법은 각 유닛의

오디오를 여러 개의 패킷으로 나누어 전송하는 것이다. 만일 패킷이 손실되면 같은 내용을 담고 있는 다른 패킷이 손실을 복구한다. 이러한 방법은 그림 3에 나타나 있다. 이 방법은 Mbone 상에서 사용하기 위하여 Hardman 등<sup>[6]</sup>과 Bolot 등<sup>[7]</sup>에 의해 제안되었으며, Podolsky 등<sup>[8]</sup>에 의해 실험되었다.

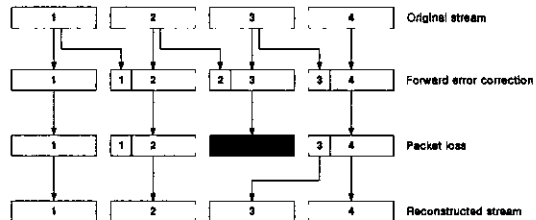


그림 3. 매체 종속 FEC를 사용한 복구

첫 번째 전송되는 오디오 데이터의 복사본은 주 인코딩이라 부르며 후에 전송되는 것을 보조 인코딩, 또는 잉여정보라 한다. 보조 오디오 인코딩에 주 오디오 인코딩과 같은 코딩 방법을 사용하는가에 대한 결정은 전송자에게 달려있지만, 일반적으로 보조 인코딩은 주 인코딩보다 작은 대역폭, 낮은 품질의 인코딩 기법을 사용한다.

인코딩의 선택은 인코딩의 계산 복잡도와 대역폭 요구에 모두 의존하는 어려운 문제이다. Hardman 등과 Bolot 등은 계산이 복잡하더라도 LPC(2.4-5.6Kb/s)나 GSM 인코딩(13.2Kb/s)과 같은 저비트율 analysis-by-synthesis 코덱의 사용이 인터넷에서 경험적으로 손실구간을 커버할 수 있음을 주장하였다.

만일 주 인코딩이 높은 처리력을 필요로 하지만 충분한 품질과 낮은 대역폭을 사용한다면 보조 인코딩은 주 인코딩과 같을 것이다. 이에 대한 예는 오늘날의 데스크 탑에서 사용가능한 처리력을 요구하며 낮은 대역폭(5.2-6.3Kb/s)을 가지는 ITU G.723.1 코덱이다.

매체 종속 FEC의 사용은 패킷 사이즈를 증가시키는 오버헤드를 발생한다. 예를 들어 보조 인코딩으로 8kHz PCM  $\mu$ -law(64kb/s)을 사용하는 경우 각 패킷의 데이터 부분 크기를 20% 증가시킨다. 매체 독립 FEC 기법과 마찬가지로 매체 종속 FEC의 오버헤드는 유용한 것이다. 어쨌든 이들 기법과 다르게 매체 종속 FEC의 오버헤드는 복구될 수 있는 손실의 수에 영향을 미치지 않고 손실을 감소시킬 수 있다. 대신 복구의 품질은 오버헤드에 의해 좌우된다. 오버헤드를 감소시키기 위해 오디오 응용에서

사용할만한 복구기법들이 사용되고 있다.

모든 패킷마다 매체 종속 FEC를 전송할 필요는 없다. 음성 신호는 80ms 이상의 묵음구간이 있다. Viswanathan 등<sup>[9]</sup>은 코덱의 파라미터가 충분히 변화할때에만 음성 유닛을 전송하는 LPC 코덱으로 동일한 품질로 30퍼센트의 대역폭을 절약할 수 있음을 설명하였다.

언급된 다른 송신자 기반 기법들과 달리 매체 종속 FEC는 하나의 패킷 지연만이 부가되는 작은 지연을 가지는 잇점이 있다. 이는 큰 종단간 지연이 허용되지 않는 상호대화 응용에 적합하다. 만일 큰 종단간 지연이 허용된다면 연속적인 손실에서 성능을 향상시키도록 패킷을 복사하여 전송할 수도 있다. 현재 시점에서 매체 종속 FEC는 많은 Mbone 음성회의의 tool을 지원한다. 매체 종속 FEC를 위한 표준 RTP의 payload도 연구되고 있다<sup>[10]</sup>.

### 2.1.2 인터리빙

인터리빙은 디지털 통신 시스템에서 연속적인 데이터 손실을 분산시키기 위해 널리 사용되는 기술이나 컴퓨터 네트워크 상에서는 지연을 야기시키는 문제 때문에 사용되기 어려웠다. 그러나 인터리빙은 다른 복구기법에 비해 전송률이나 추가적인 처리에 따른 오버헤드가 전혀 없기 때문에 지연을 줄일 수 있거나 지연이 크게 문제되지 않는 인터넷 방송 시스템 등에서는 매우 효과적인 복구기법이 될 수 있다.

어쨌든 인터리빙은 유닛사이즈가 패킷 사이즈보다 작고 종단간 지연이 중요하지 않다면 손실의 영향을 감소시키는 유용한 방법이다<sup>[11]</sup>. 유닛은 전송될 스트림에서 원래 인접한 유닛은 간격을 보장받으며 분리하기 위해 전송전에 재배열되며 수신측에서 다시 원래의 순서로 돌아온다. 인터리빙은 패킷손실의 영향을 분산시킨다. 예를 들어 유닛이 5ms의 구간을 가지고 패킷이 20ms의 구간을 가진다면 (즉, 4 유닛/패킷) 첫 번째 패킷은 1,5,9,13 유닛을 포함하며 두 번째 패킷은 2,6,10,14 유닛을 가지며 이러한 식으로 구성된다. 이는 그림 4에 잘 나타나 있다.

한 패킷의 손실은 인터리빙을 하지 않은 스트림에서 하나의 큰 gap으로 나타나는 손실구간이 인터리빙된 스트림에서는 재구성된 스트림에서 다수의 작은 gap으로 나타난다. 이렇게 손실을 분산시키는 이유는 크게 두가지가 있다. 첫째로 Mbone 오디오 툴은 일반적으로 사람 음성의 음소와 유사한 길이로 패킷을 전송한다. 따라서 하나의 패킷의 손실은 음성의 인식에 인식할 수 있을 정도로 큰 영향을

미친다. 만일 손실이 여러 음소의 작은 부분으로 나누어 진다면 이러한 손실에도 청취자가 인식하기 쉬우며 결과적으로 주어진 손실율상에서 품질을 증진하게 된다.

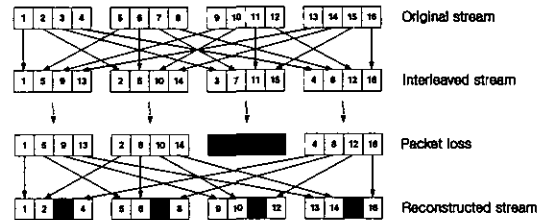


그림 4. 다수의 패킷에 대한 인터리빙 유닛

대부분의 음성과 오디오 코딩은 출력을 인터리빙할 수 있으며 인터리빙의 효율증진을 위해 수정될 수 있다. 인터리빙의 가장 큰 장점은 스트림의 대역폭 요구를 증가시키지 않는다는 것이며, 단점으로는 인터리빙의 처리로 인한 지연이 증가시킨다는 것이다.

### 2.1.3 기존 손실 패킷 복구기법의 문제점

기존의 손실 패킷 복구기법은 크게 나누어 재전송에 기반한 ARQ와 패리티 패킷 또는 잉여정보를 부가하는 FEC로 나눌 수가 있다. 이 중 ARQ는 데이터의 신뢰성은 보장할 수 있지만, 지연이 매우 크므로 실시간 서비스가 필요한 인터넷 폰에서는 사용이 어렵다.

FEC중 패리티 패킷을 생성하여 이를 패킷 복구에 이용하는 매체 독립 FEC의 경우 인코딩과 디코딩 알고리즘이 복잡하며, 잉여 정보를 부가하여 패킷 손실시 이 잉여정보를 통하여 패킷을 복구하는 매체 종속 FEC는 잉여 대역폭을 요구한다는 단점이 있다.

더 많은 패킷의 복구를 위해 잉여정보를 하나 이상을 부가하여 사용하거나 잉여정보를 부가한 후 인터리빙을 수행하는 방법<sup>[12]</sup>도 있으나 이들은 많은 대역폭을 요구하거나 지연시간이 너무 크므로 원활한 서비스 제공에는 어려움이 있다.

또한 손실 패킷의 복구를 위해서는 인터넷 상에서 패킷 손실의 특성을 먼저 분석해 볼 필요가 있다. 이는 패킷 손실의 특성에 따라 손실 복구기법의 설계에 영향을 미치므로 손실의 특성을 고려하여야 효율적인 손실 패킷 복구기법의 설계가 가능하다.

패킷 손실 중 가장 심각한 문제를 야기하는 것은 패킷이 수신측에 도착하지 않은 경우로, 네트워크

트래픽이 클수록 보다 많이 발생한다. 또한 패킷 손실은 단순히 랜덤한 분포가 아니고 트래픽이 심할수록 연속적으로 패킷을 손실할 확률도 높다. 이러한 패킷 손실 특성에 따라 연속적인 패킷 손실에 대한 복구기법이 필요하고 심각한 패킷 손실 상황에서도 서비스를 제공할 수 있는 방안이 마련되어야 한다.

### 2.2 인터리빙 FEC 기법

인터넷 상에서 손실 패킷 복구기법으로 널리 사용되고 있는 것은 FEC 기법이며 인터리빙 기법은 무선통신에서 주로 사용되고 있다. 본 논문에서는 보조 인코딩을 사용하는 매체 종속 FEC 기법에서 주 인코딩 오디오 데이터는 현재의 방식대로 사용하고 보조 인코딩 오디오 데이터, 즉 잉여정보만 인터리빙 하여 연속된 패킷 손실이나 심각한 패킷 손실 상황에서도 최소한의 서비스 품질을 유지할 수 있는 인터리빙 FEC 기법을 제안한다. 이러한 방식은 FEC와 인터리빙 기법의 장점을 모두 가지고 있다. FEC와 인터리빙 기법의 장단점을 비교하여 보이면 다음의 표 1과 같다.

표 1. FEC, 인터리빙 기법의 장단점 비교

	FEC	인터리빙
기본 개념	잉여정보를 이용하여 패킷 손실시 이를 이용	패킷의 순서를 바꿈으로써 연속된 손실에 강함
대역폭	잉여정보 만큼 요구	대역폭 증가 없음
지연	작음	큼
손실패킷 복구	바로 전 패킷만 복구가능	손실 패킷 복구는 없으나 손실의 영향을 분산시킴
실시간 응용서비스 적용	가능	인터리빙 구간 설정에 따라 다르나 일반적으로 지연이 큼

먼저 전송될 각 패킷들은 기존의 매체 종속 FEC와 마찬가지로 LPC-10 등으로 코딩된 잉여 정보를 가지게 된다. FEC의 경우, 만약 하나의 패킷이 전송중에 손실되면 바로 다음 패킷이 가지고 있는 잉여정보를 통해 이 패킷을 복구한다. 그러나 연속된 2개 이상의 패킷이 손실 될 경우 손실된 패킷 중 하나 밖에 복구할 수 없는 단점이 있다. 이는 손실되기 바로 이전의 패킷을 복사하여 사용한다 하더라도 3개 이상의 연속된 패킷 손실시에는 패킷 손실은 막을 수 없다. 이것은 FEC를 사용한 패킷은

바로 앞의 패킷정보만을 가지고 있기 때문이다.

따라서 패킷 손실이 매우 작거나 손실의 특성이 연속적이지 않은 경우에는 거의 모든 패킷을 복구할 수 있으므로 FEC 기법이 매우 유리하다 할 수 있다. 그러나 현재의 인터넷 망에서는 이러한 품질을 기대하기 어렵기 때문에 다른 대안이 필요하게 된다.

인터리빙 FEC에서는 잉여 정보를 생성할 때 이를 인터리빙 하여 전송 패킷을 생성한다. 즉, 잉여 정보가 연속되지 않고 인터리빙 되어 있으므로 연속적인 패킷 손실이 발생하여도 복구가 가능하다. 이렇게 하는 이유는 연속된 패킷 손실에서도 묶음 구간을 최소로 하여 서비스의 품질을 향상시키기 위함이다. 또한 하나의 패킷이 손실되고 이에 대한 잉여정보도 손실이 되었다면 이것은 바로 전의 패킷을 복사하여 대처하도록 하였으며, 잉여정보만 인터리빙을 수행하기 때문에 패킷과 잉여정보가 동일한 내용을 가지고 있을 수가 있으므로 이런 경우는 바로 다음 패킷의 잉여정보와 서로 교환하도록 하였다. 이를 FEC와 비교하여 표현하면 그림 5와 같다.

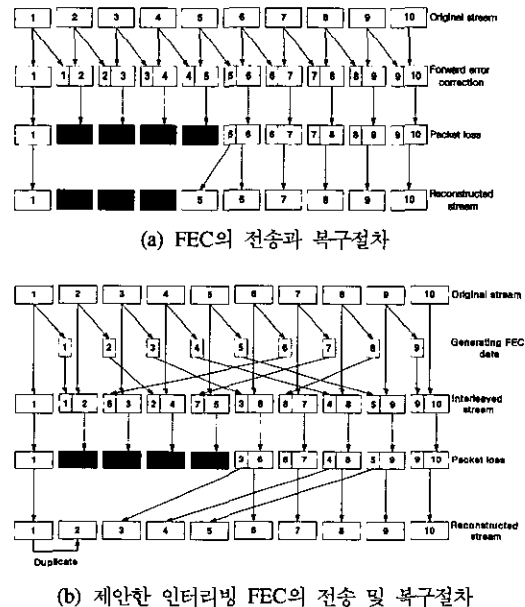


그림 5. 매체 종속 FEC와 인터리빙 FEC의 전송 및 복구절차 비교

그림 5에서 보는 바와 같이 전송 절차는 제안한 인터리빙 FEC 기법이 잉여정보를 인터리빙 하여야 하므로 FEC에 비해 알고리즘이 더 복잡하다. 그러

나 연속적인 패킷 손실에 대하여 잉여 정보를 이용한 패킷 복구가 가능하다는 장점이 있다. 특히 인터리빙 구간이 크면 클수록 복구율이 높아지는데 이는 주 인코딩 정보와 보조 인코딩 정보가 많은 시간차를 두고 전송되기 때문이다. 그러나 인터리빙 구간이 커지면 그만큼 지연도 커지므로 복구율과 지연사이에서 적당한 구간을 설정해야 한다.

인터리빙 FEC 기법에서는 인터리빙으로 인한 지연을 최소화하기 위하여 패킷 전체를 인터리빙 시키지 않고 잉여정보만 인터리빙을 시켰다. 패킷 전체를 인터리빙 시킬 경우 패킷의 처리와 인터리빙에 많은 지연이 발생하므로 성능을 향상시키고 지연을 감소시키기 위하여 잉여정보만 인터리빙하는 방식을 취한다.

즉 잉여정보를 추가한 후 패킷 전체를 인터리빙 시키는 경우 발생하는 지연은 전송중 패킷 손실의 발생에 관계없이 피할 수 없는 지연이므로 패킷 손실이 없을 시에는 전송의 효율을 저하시키는 결과를 가져오게 된다.

그러나, 본 논문에서 제안하는 바와 같이 잉여정보만 인터리빙을 할 경우 전송에 필요한 지연은 피할 수 없으나 수신후 재생시에는 패킷 손실의 여부에 따라 지연이 결정되게 된다. 즉, 패킷 전체를 인터리빙 시키는 것이 아니므로 주 인코딩 정보는 원래의 순서에 따라 재생하면 되고 패킷 손실이 발생할 시에만 보조 인코딩 정보를 디인터리빙하여 복구해야한다. 따라서 패킷의 손실이 없을 경우에는 수신단에서 지연이 발생하지 않는다. 그러나 패킷 손실이 심할 경우 계속해서 보조 인코딩 정보를 가져와야 하므로 지연시간이 길어진다.

예를 들어 같은 2x5 인터리빙 구간을 가지는 인터리빙 FEC 기법과 패킷 전체를 인터리빙 시키는 방법을 고려해보면 패킷 손실이 없는 경우의 지연 시간은 다음과 같이 구할 수 있다.

① 송신측의 지연

- 인터리빙 FEC

$$D_s = m \times n - (m + n) + 1 = 2 \times 5 - (2 + 5) + 1 = 4$$

- 잉여정보 추가후 패킷 인터리빙

$$D_s = 2 \times 5 - (2 + 5) + 1 = 4$$

② 수신측의 지연

- 인터리빙 FEC

$$D_R = 0$$

- 잉여정보 추가후 패킷 인터리빙

$$D_R = D_s + m \times n = 4 + 2 \times 5 = 14$$

즉, 이상에서 보는 바와 같이 인터리빙 FEC는 송신측에서 4 패킷의 지연이 발생하지만 수신측에서는 지연이 없고, 패킷 전체를 인터리빙 시키는 방식은 송신측에서 4 패킷, 수신측에서 14 패킷의 지연이 발생하게 되므로 잉여정보만 인터리빙하는 인터리빙 FEC가 지연이 훨씬 작은 것을 알 수 있다.

다음의 그림 6은 제안한 인터리빙 FEC 방식의 흐름도를 나타낸다. 흐름도에서 보는 바와 같이 입력된 음성 데이터는 두가지 인코딩 방법을 통하여 주 인코딩과 보조 인코딩으로 생성된다. 이중 보조 인코딩은 인터리빙을 통하여 순서가 바뀌게 되고 다시 주 인코딩에 추가되어 패킷으로 전송된다. 이때 만일 보조 인코딩과 주 인코딩의 내용이 같으면 바로 다음 패킷의 보조 인코딩과 교환을 하여 하나의 패킷에 같은 내용을 가지고 있는 것을 방지한다.

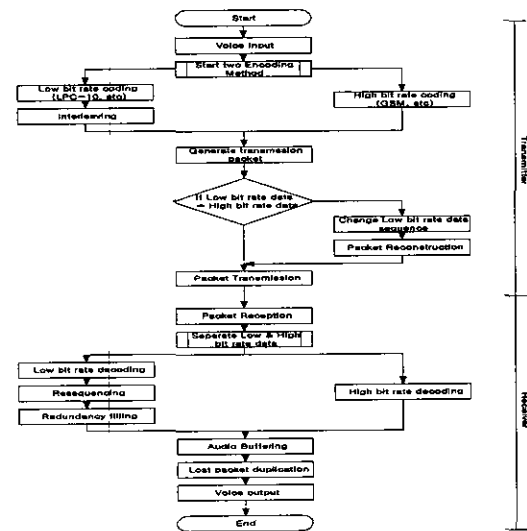


그림 6. 제안 인터리빙 FEC 방식의 흐름도

이렇게 전송된 패킷은 수신측에 수신되어 다시 주 인코딩과 보조 인코딩으로 나누어져 디코딩되며 보조 인코딩 데이터는 다시 디인터리빙을 하여 원래의 순서대로 바꾸어 준다. 만일 주 인코딩의 데이터가 손실되었을 경우 보조 인코딩 데이터로 채우고 오디오 버퍼링을 한 후, 이 때에도 복구가 안된 부분은 바로 전의 패킷을 복사하여 채움으로써 최소한의 서비스 품질을 보장할 수 있도록 한다.

### III. 실험 및 결과분석

#### 3.1 시뮬레이션 환경

본 논문에서 제안한 인터리빙 FEC의 성능을 평가하기 위해 FEC 기법 및 인터리빙기법과의 성능을 비교하였다. 성능비교는 랜덤한 패킷 손실에 대한 패킷 복구율, 연속된 패킷 손실에 대한 패킷 복구율을 분석하였으며 각각을 FEC와 인터리빙 기법과 비교하여 그래프로 나타내었다.

인터넷 폰 서비스의 손실패킷 복구를 위한 인터리빙 FEC의 적용을 위하여 소켓프로그램을 응용한 클라이언트 서버 프로그램을 작성하였다. 서버프로그램에서 정형화된 패킷을 입력받아 전송하며, 랜덤 함수를 통한 임의의 패킷에 영향을 주어서 클라이언트 프로그램에서 랜덤한 패킷 손실에 대한 패킷 복구율, 연속된 패킷 손실에 대한 패킷 복구율을 분석하게 된다.

패킷을 전송하고 수신하는 과정에서 FEC방식, 인터리빙방식, 인터리빙 FEC방식을 적용하여 손실을 주기 위한 패킷을 랜덤하게 설정하기 위해서 랜덤 함수(Packet\_Loss = Random())을 사용하였다.

이러한 일련의 실험들에 사용된 인터리빙 FEC방식은 다음과 같은 프로그래밍 과정으로 작성되었다.

▶ Server 프로그램과정

```
void main()
{
- ClientReceive();//클라이언트에서 송신된 패킷받음
- RecoverAnalysis();// 도착한 패킷의 분석
- RecoverPacket(IpIP, IpRP, nPACKET); //패킷복구
- OriginalData();//원래패킷으로 복구
- ClientSend();// 클라이언트로 분석결과 송신
}
```

▶ 클라이언트 프로그램과정

```
typedef enum {FALSE, TRUE} BOOL;
typedef struct tagPacket { // 이용할 포맷의 정의
    BOOL IsInvalid;
    int Head;
    int Tail;
} PACKET, *LPPACKET;

void main()
{
-패킷수 입력받음
-패킷손실을 입력받음
-InterleavePacket(IpSP, IpIP, nPACKET); //인터리빙함수 호출
-//DoRPLoss(IpIP, nLOSSsPACKET, nPACKET);
```

```
//랜덤한경우패킷손실
-DoCPLoss(IpIP, nLOSSsPACKET, nPACKET);
// 연속적인 경우 패킷손실
- ServerSend() //서버에 패킷송신
- RevResult() // 서버에서 송신된 패킷분석결과 받음
}
```

▶인터리빙 함수 과정

```
void RecoverPacket(RecoverPacket(IpIP, IpRP, nPACKET);
{
for( int i=1; i<=nPacket; I++) {
if(패킷의 상태가 FALSE상태라면...) {
pRP[pIP[i].Head].Flag = FALSE;
pRP[pIP[i].Head].Tail = pIP[i].Head;
pRP[pIP[i].Tail].Flag = FALSE;
pRP[pIP[i].Tail].Tail = pIP[i].Tail;
}
}
```

랜덤함수를 통한 손실패킷이 정해지면 그 패킷을 인터리빙 FEC을 이용하여 원래 인덱스를 가진 순서대로 정렬함수(SORT())을 이용하여 정렬하고 빈 인덱스를 검색함수(SER())을 통해서 빠진 인덱스를 파악하여 어떤 패킷이 손실되었는지를 검색하게 된다. 검색후에는 복원률 산출 함수(R\_Rate())을 통해서 복원율을 산출하게 된다.

이러한 일련의 실험들은 클라이언트 프로그램을 설치한 PC와 서버프로그램을 설치한 PC에서 실험되었으며, 패킷에 영향을 줄 수 있는 환경은 트래픽에 따른 부하에 따라서만 변화하도록 하였다.

또한 이 방식의 비교를 위해서 FEC 방식 및 인터리빙을 같은 방식으로 시뮬레이션하여 손실패킷 복구율의 비교분석 자료로 활용하였다.

3.2 손실패킷 복구기법 비교실험 결과분석

그림 7은 랜덤한 패킷 손실을 발생시키는 서버/클라이언트 시뮬레이션 프로그램을 이용하여 손실률의 변화에 따른 복구율을 측정된 결과로서, 패킷 손실에 따른 각 방식별 복구율을 구해본 것이다.

그림 7을 보면 패킷손실이 전체패킷의 10%, 20%,... 50%증가함에 따라 손실된 패킷에 대한 복구율은 기존의 FEC방식보다 인터리빙 FEC방식이 평균 4.81% 향상됨을 알 수 있다.

그림 8은 연속적인 패킷 손실에 대한 패킷 복구율을 분석한 것으로서 패킷을 10%, 20%, ..., 50% 까지 영향을 주어 연속적인 패킷손실이 있더라도

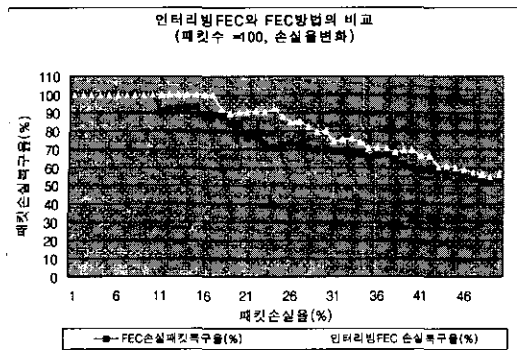


그림 7. 랜덤한 패킷손실에 대한 패킷 복구율 분석

복구율은 랜덤한 패킷 손실과 같이 복구율이 인터리빙 방법보다는 5%정도의 손실율까지는 비슷한 패킷복구율을 나타내지만 그 이후로는 2%정도 향상됨을 알 수 있다.

FEC 기법보다는 처음부터 약 2.5%의 차이로 인터리빙 FEC 가 복구율이 향상됨을 알 수가 있다.

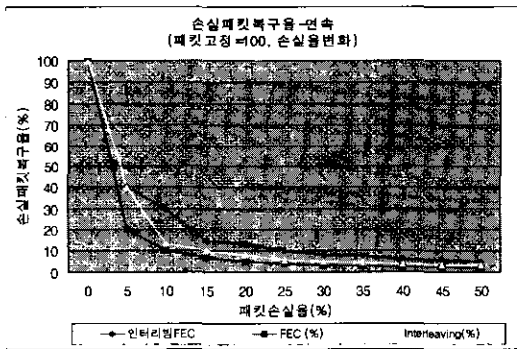


그림 8. 연속적인 패킷 손실에 대한 패킷 복구율 분석

패킷 손실이 연속적인 경우, 즉, 손실 특성이 버스트한 경우에는 FEC는 다음 패킷이 가지고 있는 잉여정보만을 가지고 복구를 수행하므로 바로 전의 패킷은 복구할 수 있으나 2개 이상 연속된 패킷 손실은 복구할 수 없다. 이에 반해 인터리빙 FEC는 잉여정보가 분산되어 있으므로 수신단에서 재생중 손실 패킷이 있다면 잉여정보들을 디인터리빙시킨 후 복구할 확률이 높아지게 된다.

FEC 기법, 인터리빙 기법, 제한한 인터리빙 FEC 기법의 지연시간을 살펴보면 다음과 같다. 여기서 지연시간은 망에서의 지연 등은 제외한 패킷 생성과 패킷 수신후 재조립 과정에만 소요되는 지연시간을 의미하는 것으로 FEC의 경우 인터리빙이 없으나 수신단에서 패킷이 손실 될 경우 1패킷의 지

연이 발생한다. 만일 한 패킷이 20ms의 구간을 가진다고 가정하고 인터리빙을 사용하는 경우 2x5로 구간을 설정하였을 때, 패킷 손실율의 변화에 따른 패킷당 평균지연시간을 분석한 결과를 보이면 다음의 그림 9와 같다.

그림 9에서 보는 바와 같이 인터리빙은 패킷 손실과는 상관없이 항상 패킷마다 360ms의 지연이 발생한다. 이는 인터리빙에도 지연이 발생하지만 음성의 재생이 반드시 디인터리빙후에만 가능하기 때문이다. 이에 반해 인터리빙 FEC의 경우 패킷 손실의 정도에 따라 지연시간이 변화하며 인터리빙 기법에 비해서는 매우 낮은 지연을 보임을 알 수 있다.

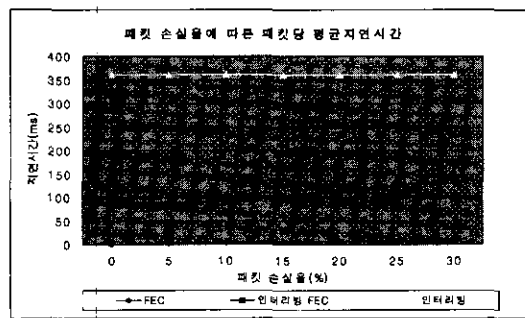


그림 9. 패킷 손실율에 따른 지연시간 비교

또한 인터리빙 FEC 기법에서는 인터리빙 구간에 따른 복구율에 변화가 있다. 이는 일반적으로 구간이 클수록 높은 복구율을 나타낸다. 이에 대한 실험 결과는 표 2, 표 3에서 보는 바와 같다.

표 2, 표 3은 인터리빙 구간, 즉 M과 N을 변화시키면서 패킷 손실이 랜덤한 경우, 연속인경우에 대하여 패킷 손실율이 20%일때의 복구율을 나타낸 것이다. 표 2, 표 3에서 보는 바와 같이 패킷 복구율은 M과 N이 커질수록 증가하며, 표에서 음영으로 표시한 것과 같이 대각선 방향으로 유사한 값을 가진다.

표 2. 인터리빙 구간에 따른 연속 손실패킷 복구율 (패킷손실율: 20%)

	2	3	4	5	6	7	8	9	10
2	6.25	12.50	16.67	18.18	21.43	22.22	25.00	33.33	41.67
3	9.36	12.57	16.33	17.33	21.50	23.11	32.86	41.62	43.00
4	12.50	15.63	16.67	17.86	22.22	24.00	41.67	50.00	52.63
5	14.50	12.50	15.63	17.86	25.00	41.67	42.50	45.00	45.83
6	16.18	21.43	22.22	28.57	37.50	39.47	44.19	47.37	54.17
7	18.86	21.18	24.18	36.00	41.90	50.00	50.00	52.00	55.36
8	19.63	28.00	39.47	42.11	50.00	50.00	50.98	53.13	54.39
9	24.86	34.43	39.00	46.51	47.22	47.62	50.00	50.88	54.00
10	31.50	36.63	41.67	46.50	47.25	48.61	51.56	56.25	57.14



표 3. 인터리빙 구간에 따른 랜덤 손실패킷 복구율 (패킷손실율: 20%)

↖	2	3	4	5	6	7	8	9	10
2	84.56	85.00	84.00	85.00	87.78	88.50	90.50	92.33	96.86
3	85.22	84.42	85.00	88.95	88.00	90.56	92.95	96.25	97.26
4	84.33	85.68	88.13	88.43	89.36	91.84	95.50	97.72	97.50
5	85.45	87.00	88.29	89.55	92.14	95.62	97.50	97.88	97.91
6	87.67	87.68	88.00	91.95	94.17	96.82	97.42	97.71	98.05
7	88.00	88.00	92.00	95.00	96.78	97.17	97.25	98.38	97.86
8	88.67	91.42	94.63	96.57	97.25	97.82	98.33	98.67	98.70
9	91.56	94.68	97.00	97.00	97.34	98.33	98.71	98.86	98.76
10	94.64	96.67	96.66	98.00	97.70	98.57	98.25	99.00	99.47

또한 랜덤한 패킷 손실보다는 연속적인 패킷 손실이 인터리빙 구간에 따라 복구율에 많은 차이를 보인다.

인터리빙의 경우 지연시간은 인터리빙 구간  $M \times N$ 에 따라 결정되며 패킷 손실율을 10%로 가정하고  $M$ 과  $N$ 값을 변화시켰을 때 인터리빙과 인터리빙 FEC 기법의 지연시간을 보이면 다음의 그림 10과 같다.

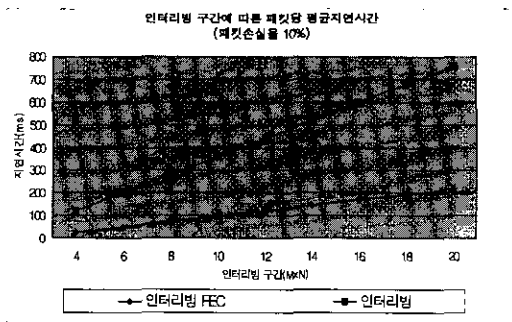


그림 10. 인터리빙 구간에 따른 지연시간 비교

인터리빙에 구간에 따른 지연시간은 구간이 클수록 지연이 커지는 양상을 보인다. 따라서 사용하고 자 하는 어플리케이션에 따라 인터리빙 구간의 조정이 필요하다. 즉, 본 논문에서는 실시간 서비스인 인터넷 폰을 대상으로 하였기 때문에 지연에 한계가 있으므로 작은 지연으로 적절한 복구율을 얻을 수 있는  $2 \times 5$ 의 인터리빙 구간을 사용하였으나, 인터넷 방송과 같이 약간의 지연이 허용되는 서비스라면 지연이 좀 더 크더라도 손실 패킷 복구율을 높일 수 있도록 인터리빙 구간을 크게 설정해주는 것이 유리할 것이다.

이상의 시뮬레이션 결과에서 본 바와 같이 인터리빙 FEC는 기존의 FEC에 비해 손실 패킷의 복구율이 높음을 알 수 있다. 또한 지연의 경우 인터리

빙 기법은 실시간 서비스가 어려운 정도의 지연을 나타내지만 인터리빙 FEC는 패킷 손실의 정도에 따라 다르지만 인터리빙에 비해서는 매우 낮은 지연을 보였다. 이러한 결과로 볼 때 제안한 인터리빙 FEC는 실시간 음성 서비스인 인터넷 폰에서 충분히 사용이 가능하다.

또한 인터리빙 구간의 설정이 중요한데 본 논문에서는 인터넷 폰에 중점을 두고 시뮬레이션 하였으므로  $2 \times 5$ 를 사용하였으나 다른 오디오 서비스의 경우에는 지연과 손실 패킷 복구율이 적절한 수준을 유지하도록 설정하면 되겠다.

#### IV. 결론

본 논문에서는 인터넷 폰 서비스를 제공하는 데 중요한 요소기술중 하나인 손실 패킷 복구를 위한 인터리빙 FEC 기법을 제안하였다. 인터리빙 FEC는 기존에 패킷손실 복구에 사용되었던 잉여정보를 가지는 매체 중속 FEC 기법과 무선통신에서 주로 사용되었던 인터리빙 기법을 사용하였다.

시뮬레이션 결과 제안한 인터리빙 FEC 모형은 랜덤한 손실이나 연속적인 손실의 경우 모두 기존의 FEC에 비해 약 2~2.5% 정도 복구율의 향상이 있었다. 랜덤한 패킷 손실의 경우 현재 인터넷 폰에서 널리 사용되고 있는 FEC와 비교해 볼 때 거의 비슷한 복구율을 보였다. 연속된 패킷 손실의 경우에는 FEC에 비해 인터리빙 FEC가 높은 복구율을 보인다.

패킷 손실율에 따른 패킷당 평균지연시간은 패킷 손실이 커질수록 FEC와 인터리빙 FEC는 지연시간이 커지지만 인터리빙은 지연시간에 변화가 없다. 그러나 인터리빙은 패킷 손실이 없을 경우에도 매우 큰 지연시간을 보이므로 실시간 서비스에 적용하기는 어렵다.

인터리빙 구간의 변화에 따른 손실 패킷 복구율과 지연시간은 인터리빙 구간이 증가할수록 패킷 복구율은 증가하지만 지연시간도 커짐을 알 수 있다. 이는 인터리빙 구간이 크면 클수록 패킷 손실시 이를 복구하기 위해 더 많은 패킷이 도착하기를 기다려야 하기 때문이다. 따라서 적용하고자 하는 서비스의 특성에 따라 적절한 인터리빙 구간을 결정하는 것은 매우 중요한 문제라 할 수 있다.

이러한 시뮬레이션 결과를 통하여 인터리빙 FEC는 FEC보다는 약간 큰 지연으로 높은 복구율을 얻을 수 있음이 확인되었다. 인터리빙 FEC는 실시간

