

# IMT-2000 버스트 에러 환경에서 MPEG-4 비디오 전송에 관한 연구

정회원 서덕영\*, 홍인기\*, 유인태\*

## A Study on MPEG-4 Video Transmission on Burst Loss Environments of IMT-2000

Doug Young Suh\*, Een Kee Hong\*, In Tae Ryoo\* *Regular Members*

### 요 약

이동 통신 환경에서는 비트 에러가 버스트 형태로 일어난다. 본 논문은 이러한 환경에서 MPEG-4 비디오를 전송하였을 때, 영향을 분석한다. MPEG-4에서 제공되는 에러 강인성 도구들이 얼마나 효과적인가를 에러의 버스트 정도에 따라 얼마나 효과적인가를 확률론적으로 해석하고, 컴퓨터 시뮬레이션 결과와 비교 분석한다. 이 분석 결과에 따라 버스트 에러의 정도에 따라 에러 강인성 도구들을 효과적으로 운용하는 방법을 정량적으로 제시한다.

### ABSTRACT

In the IMT-2000 channel, bit errors occur in bursty fashion. This paper analyzes the effect of bursty error on transmitted MPEG-4 video. It includes probabilistic analysis of effectiveness of the MPEG-4 error resilience tools with respect to extent of error burstness. The results is compared to those of computer simulation in which corrupted bit stream is decoded with and without the MPEG-4 error resilience tools. The results can be a guideline for error resilience tools, and determine adaptive parameters for time varying channel condition.

### I. 서론

“Any where, any time” 정보를 전달하는 것을 목표로 정보통신 기술은 날로 발전하고 있다. 특히, 광대역 무선통신은 이 목표에 가장 부합되는 서비스이다. 2000년 후반부터 우리 나라를 비롯한 여러 나라에서 상용 서비스가 시작되는 IMT-2000은 향후 384kbps, 2Mbps까지 고속으로 서비스 될 예정이다. 음성은 압축하면, 약 10kbps 정도가 되며, CD 품질 오디오는 120kbps정도된다. 따라서, IMT-2000을 음성이나 오디오 전송에만 이용한다면 대역폭이 낭비되는 것으로 볼 수 있다. IMT-2000은 비디오까지 포함한 본격적인 멀티미디어 정보를 any where, any time으로 전송할 수 있다.

21세기가 되면서 비디오 서비스는 더욱 중요해지고 있다. 이제 구매력을 가지기 시작하는 세대들은 문자보다는 영상을 통하여 정보를 얻는데 더욱 익숙하며, 게임이나 영화 등 영상정보를 이용하는 오락에 더욱 관심이 크기 때문이다.

비디오 서비스 방법은 VOD 같은 스트리밍(streaming) 서비스와 화상전화 같은 대화형 서비스로 나눌 수 있다. 두 서비스 모두 실시간 서비스이므로 지연과 지터에 민감하다. 또한, 압축된 비디오에서는 상관성을 최소화 하였으므로 에러가 생기면 한 비트의 에러라도 여러 프레임의 영상의 화질을 크게 떨어뜨릴 수 있다. 음성과 오디오는 10ms 또는 20ms짜리 프레임들이 서로 독립적으로 디코딩될 수 있어서 해당 프레임만 버리고 보간하면 큰 문제

\* 경희대학교 전자공학과 (suh@khu.ac.kr)

논문번호 : K01024-0118, 접수일자 : 2001년 1월 18일

※ 본 연구는 정보통신 우수대학원 지원사업 및 ETRI(방송미디어 연구부) 지원으로 수행되었습니다.

가 없다<sup>[1][2]</sup>. 그러나, 비디오는 움직임 보상을 이용하므로 보통 0.5초 동안의 데이터가 서로 관련을 가지기 때문에 에러에 매우 민감하다.

MPEG-4 비디오 코덱 [3]의 한 특징은 이러한 에러에 대한 대책을 여러 가지 구비하고 있는 것이다. 에러의 전파를 막는 재동기 마커 (RM: resynchronization marker), 움직임 벡터와 DCT 계수를 분리하는 데이터 분리 (DP: data partitioning), 양쪽으로 디코딩이 가능한 가변 코딩 (RVLC: reversible VLC) 등이다. 이러한 도구들은 이전에 MPEG-2 [4]나 H.263 에서는 없던 기능으로써 에러가 있는 환경에서 잘 동작한다. 그러나, 이러한 기능을 사용하면 전송효율이 떨어지고 코덱의 복잡성이 증가한다. 또한, 각 기능을 어느 정도로 사용하는가하는 동작인자를 결정하는 것은 실제 통신망 환경에 따라야 하는 것이다.

본 논문에서는 IMT-2000의 통신망 환경에서 각 에러 대비 도구들이 얼마나 효과적인가, 그리고 그 도구들을 효과적으로 사용하려면 동작 인자를 어떻게 정해야 하는지를 연구하였다. Suh [5]는 에러가 랜덤하게 발생하는 것으로 보고 분석하였다. 그러나, 여러 개의 다중 경로로 구성되는 무선 채널의 경우, 채널 특성은 각 경로의 벡터적인 합으로 결정되어진다. 이때, 결합되는 각 경로의 위상에 따라 전체 신호의 크기가 커지기도 하고 작아지기도 하며 이러한 특성은 시간적으로 일정 구간 동안 유지하는 성질이 있다. 따라서 일정 기간 동안 신호의 크기가 일정 수준보다 적게 수신될 경우 버스트 에러를 나타내게 된다<sup>[6]</sup>.

이러한 무선 채널의 버스트한 특성은 인터리빙을 통하여 어느 정도 극복할 수 있으나, 인터리빙에 의한 시간 지연으로 인하여 완전히 랜덤화 할 수는 없다.

본 논문에서는 2장에서 버스트 에러가 일어나는 랜덤한 패턴을 2-state Gilbert 모델로 모델링하여 중요한 인자들의 상관관계를 유도한다. 3장에서는 이 Gilbert 모델에 따라 구해진 버스트 길이의 확률 밀도 함수를 적용하여, RM, DP, RVLC의 효과를 분석하는 수식을 유도한다. 4장에서는 유도된 수식으로 계산한 결과와 실제 MPEG-4로 인코딩한 비트스트림에 버스트 에러를 주입하여 다시 디코딩한 결과를 비교한다. 두 가지 결과가 매우 유사한 것을 알 수 있었으며, 버스트 에러 환경에서는 RVLC는 효과가 없으며, RM이 효과적임을 알 수 있었다. 5장에서 이러한 결과를 종합하여 기술한다.

## II. 배경

### 2.1 버스트 에러에 대한 Gilbert 모델

MPEG-4 표준을 정할 때 W-CDMA 채널의 에러를 시뮬레이션 하는 프로그램을 이용하였으며<sup>[10]</sup> 본 논문에서도 이 패턴을 이용하기로 한다. 이 시뮬레이션 프로그램은 두 가지 종류의 에러 패턴을 발생시킨다. 하나는 비트 에러율이 랜덤하게 분포되는 경우이고 다른 하나는 Gilbert 모델에 따른 2 상태 모델이다. 여기서, 에러는 memory-less 시스템에서 발생하며, GOOD과 BAD 두 가지 상태가 존재한다. 각 상태의 상태 전이 확률이 주어지고, 상태마다 평균 BER(bit error rate)가 주어진다.

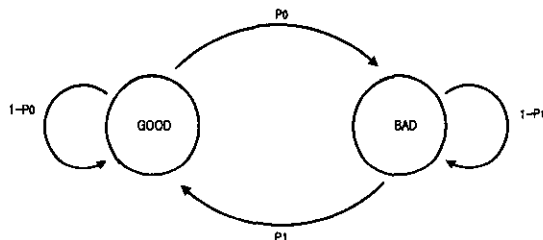


그림 1. Gilbert 모델 (각 상태에서의 전이 확률과 BER로 표현된다.)

매 비트마다 0과 1사이에서 랜덤 숫자를 발생시켜서 GOOD에서는  $p_0$  이하이면, 또한 BAD에서는  $p_1$  이하이면 상태가 전이 된다. 해당 비트가 0인가 1인가 하는 것은 BER과 관계있다. 상태가 결정되면 다시 0과 1사이에서 랜덤숫자를 발생시켜서 BER 이하이면 1, 이상이면 0을 출력한다. 1이 출력되는 것은 에러가 났음을 의미한다. 전이될 확률이  $1/n$ 인 상태의 평균 길이는  $n$ 이므로, GOOD 상태와 BAD 상태의 평균길이는 각각  $\bar{l}_g = 1/p_0$ ,  $\bar{l}_b = 1/p_1$ 이 된다.

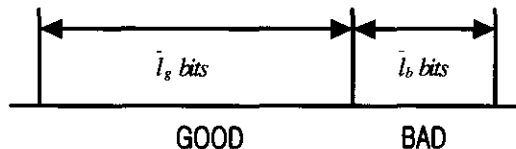


그림 2. 각 상태의 평균길이는 전이 확률의 역수이다.

GOOD의 평균길이  $\bar{l}_g$  는  $p_0$ 의 역수이며 다음과 같다. 전체 평균 BER을  $p_e$ 라고 하면,

$p_a = \frac{\overline{l_b} p_b + \overline{l_g} p_g}{\overline{l_g} + \overline{l_b}}$  이므로,  $\overline{l_g} = \frac{\overline{l_b}(p_b - p_a)}{p_a - p_g}$  이고, 전체에서 BAD 상태가 차지하는 비율은 다음과 같다. 이것이 에러 강인성 도구로 다다를 수 있는 손실율의 한계이다.

$$\text{LossLimit} = \frac{\overline{l_b}}{\overline{l_b} + \overline{l_g}} = \frac{p_a - p_g}{p_b - p_g} \quad (2.1)$$

따라서,  $p_0 = \frac{1}{l_g} = \frac{p_1(p_a - p_g)}{p_b - p_a}$  이 된다.

랜덤한 에러는 모두 FEC를 이용하여 정정된다면, 잘못 정정되었거나, 정정되지 못하고 에러 검출만 된 프레임은 버스트 에러로 간주할 수 있다. 이 경우에는 버스트 에러의 길이가 한 프레임의 길이가 되므로 일정하다고 할 수 있고, 한 프레임이 pass할 것이나 fail할 것이냐만 결정하면 된다. W-CDMA에서 프레임의 길이는 대개 10 ms 또는 20 ms이다.

버스트 길이의 평균은  $\overline{l_b} = \frac{1}{p_1}$  이다. 버스트 길이의 확률 분포는 다음과 같다.

버스트 길이가 n일 확률은 다음과 같다.

$$\Pr(l_b = n) = (1 - p_1)^{n-1} p_1 \quad \text{for } n=1,2,3 \quad (2.2)$$

그러면, 길이가 N인 비트열에 발생하는 평균 버스트 에러의 개수는 아래와 같다.

$$\text{Hypothesis} : \frac{N}{\overline{l_g} + \overline{l_b}} \quad (2.3)$$

위 식은  $N \gg \overline{l_g} + \overline{l_b}$ 인 경우에 의미가 있다. 그러면,  $N \ll \overline{l_g} + \overline{l_b}$ 이고,  $N > \overline{l_b}$ 일 때, 해당 segment가 버스트에러를 만날 확률은 다음과 같이 구할 수 있다.

$$\Pr\{N \text{과 } \overline{l_b} \text{가 겹친다.}\} = N \cdot \frac{\overline{l_b}}{\overline{l_g} + \overline{l_b}}$$

## 2.2 MPEG-4 에러 강인성 도구 (MPEG-4 error resilience tools)

MPEG-4에서 에러 강인성을 위한 방법은 에러의 전파를 막는 재동기 마커 (RM: resynchronization marker), 움직임 벡터와 DCT 계수를 분리하는 데이터 분리 (DP: data partitioning), 양쪽으로 디코딩이 가능한 가변 코딩 (RVLC: reversible VLC), 중

요한 데이터가 있는 헤더를 특별히 보호하는 헤더 에러 정정 (HEC: header error correction) 등이다.

### 2.2.1 RM (재동기 마커)

MPEG-4에서는 대부분의 변수를 가변비트율 코딩 (VLC: variable length coding) 방법으로 코딩하기 때문에 한 번 에러가 발생하면 계속적으로 그 영향이 전파된다. 에러의 전파를 막기위하여 VOP 내의 비트스트림에 일정한 간격으로 RM을 삽입함으로써 비트스트림에 에러가 발생되더라도 그 패킷 (RM과 다음 RM간의 비트스트림)만을 버려서 오류가 확산되는 것을 막을 수 있다. RM으로 구분되는 일련의 비트열을 MPEG-4에서는 비디오 패킷이라고 정의한다. RM은 17비트의 비트 패턴이다. 비디오 패킷의 길이는 표준으로 정하지 않았으나 3GPP에서는<sup>[9]</sup> 대개 하나의 비디오 프레임(한 화면)당 4-5 개 정도가 있는 것이 권고된다. 비디오 패킷간 서로 독립적으로 디코딩이 가능하여야 하며 각 비디오 패킷은 시작하는 매크로 블록의 주소를 포함하고 있다. 비디오 패킷의 최소 크기는 인코더 디코더간 약속해야 하는 사항이며, 최소 크기를 넘은 후에는 새로운 매크로 블록 데이터를 시작할 수 없다.



그림 3. 비디오 패킷의 구조 (RM으로 시작해서 다음 RM전까지 하나의 비디오 패킷이며, 여러 개의 매크로 블록 데이터가 포함된다.)

### 2.2.2 RVLC (양방향 가변 비트율 코딩)

엔트로피 코딩의 코드를 만들 때, 역방향으로도 디코딩이 가능하도록 한다. 그러면, 비트스트림에 에러가 발생하였을 때 순방향으로 디코딩하다가 에러가 포착되면, 다음 RM을 찾아 역방향으로 디코딩이 가능하게 한다. 따라서 버려지는 비트수를 줄일 수 있게 된다. MPEG-4에서는 DCT 계수뿐만 아니라, 움직임 벡터에 대해서도 매우 효율적인 RVLC용 코드북을 제공한다. RVLC를 사용에 따른 압축율의 저하는 미미하다.

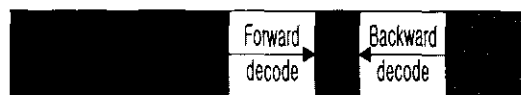


그림 4. RVLC (순방향으로 디코딩하다가 에러를 만나면, 다음 RM을 찾고 역방향으로 디코딩한다.)

2.2.3 DP (데이터 분리)

MPEG-4에서 움직임 벡터 정보는 DCT 정보보다 짧으나, 더 중요한 정보를 가지고 있다. 이 두 종류의 정보를 분리하여 DCT 데이터에서 생긴 에러가 움직임 벡터 정보에 영향을 주지 않도록 하는 방법이다. 따라서, 움직임 벡터 정보에서는 에러가 생기지 않고, DCT 계수에서는 에러가 생기면 움직임 벡터 정보만 가지고 디코딩을 하며, 이 경우에도 화질저하는 참을 수 있는 정도가 된다. 움직임 벡터 정보와 DCT 데이터 사이는 MM (motion marker)를 두어 재동기를 할 수 있도록 한다.



그림 5. DP(한 비디오 패킷내 모든 매크로 블록의 움직임 벡터를 먼저 싣고, MM으로 분리한 후, DCT 데이터를 넣는다.)

2.2.4 HEC (헤더 에러 정정)

하나의 VOP (대개, 영상 한 프레임)의 헤더는 중요한 정보를 가지고 있다. VOP 헤더가 손상되면 그 VOP는 버려야 한다. 이 중요한 정보를 보호하기 위하여 각 비디오 패킷의 헤더에 헤더정보를 반복하여 싣는다. 헤더 정보는 매우 짧기 때문에 반복에 의한 비트율 증가는 미미하다.

Ⅲ. MPEG-4 에러 강인성 도구의 효과에 대한 확률적 분석

버스트 에러가 생기는 패턴에 대한 분석을 토대로 MPEG-4 비디오 스트림에 버스트에러가 유입되었을 때의 영향을 분석한다. 버스트 에러의 영향에 따라 RM, DP, RVLC, HEC와 같은 MPEG-4 에러 강인성 (error resilience) 도구들의 개별적인 성능을 분석할 수 있다.

3.1 영상 한 프레임당 버스트 에러 출현 횟수

버스트 에러의 평균 길이는 3GPP에서 전형적으로 가정하는 값으로 가정한다. 즉,  $t_b[\text{sec}] = 1\text{ms}, 10\text{ms}, 20\text{ms}$  등이다. 비디오의 초당 화면수를  $\text{fps}[\text{sec}^{-1}]$ , 즉 frames per seconds이라고 하면, 한 frame 길이는  $1/\text{fps}[\text{sec}]$ 이다. 채널의 전송속도를  $r$  bps라고 하면, 한 프레임에 나타나는 평균 버스트 에러의 수  $N_b$ 는  $N_b = \frac{r/\text{fps}}{t_b + t_g}$ 이며, (2.1), (2.3) 식

을 대입하면,

$$N_b = \frac{r/\text{fps}}{r \cdot t_b \cdot (1 + \frac{p_b - p_a}{p_a - p_g})} = \frac{p_a - p_g}{\text{fps} \cdot t_b \cdot (p_b - p_g)}$$

된다.

3.2 비디오 패킷의 크기와 버스트 에러의 상관성

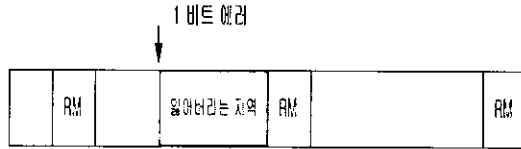


그림 6. 비트 에러 영향의 전파 (비트 에러가 하나만 발생하여도 다음 RM까지 모든 정보는 버려야 한다.)

MPEG-4에서는 비디오 정보를 가변비트율로 코딩을 하게 된다. 따라서, 한 비트 에러가 발생하더라도 그 영향이 다시 동기화될 때까지 계속된다. 보통은 하나의 VOP (하나의 영상 프레임) 헤더에서 동기화된다. 그러나 RM을 사용하면, 비디오 패킷헤더에서 재동기 될 수 있다. 비디오 패킷 헤더의 맨 앞에 RM이 놓인다. 3GPP에서는 한 VOP당 4-5개의 비디오 패킷을 넣도록 권고하고 있으나<sup>[9]</sup>, 이에 대해서는 상황에 따라 사용자가 결정할 수 있게 되어 있다. RVLC를 이용하면, 비디오 패킷의 마지막 부분부터 상당한 부분을 살릴 수 있다.

3.2 사용하는 도구와 상관없이 공통적인 손실,  $loss_{common}$

헤더가 손상되면, 이에 속하는 데이터는 사용하는 다른 도구와 관계없이 모두 잃어버린다. 본 절에서는 도구의 사용과 관련없이 공통적으로 나타나는 손실을 다룬다. 본 절에 나오는 수식을 유도하는데는 다음 두 식이 유용하게 이용된다.

$$\sum_{n_1}^{n_2} r^n = \frac{r^{n_1} - r^{n_2+1}}{1 - r}$$

$$\sum_{n_1}^{n_2} nr^n = \frac{1}{1-r} (n_1 r^{n_1} - n_2 r^{n_2+1} + \frac{r^{n_1+1} - r^{n_2+1}}{1-r})$$

공통적인 손실은 3가지 경우로 나누어서 분석할 수 있으며, 각각의 경우는 독립적이므로 그 영향을 합할 수 있다.

$$loss_{common} = loss_{2nd} + loss_{Header} + loss_{VOPHeader}$$

위의 식 우변의 세 항은 다음 절에서 설명한다.

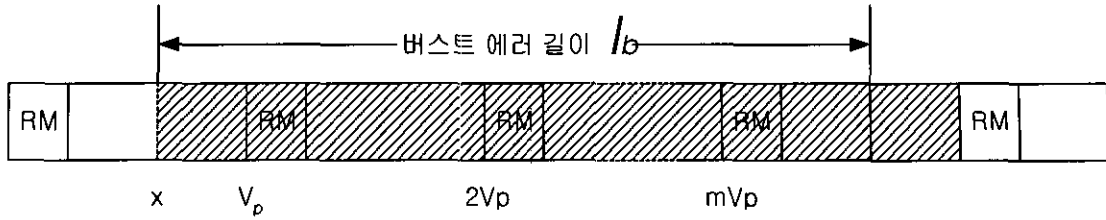


그림 7. 버스트 에러가 두 개 이상의 비디오 패킷에 걸치는 경우 (두 번째부터는 패킷 헤더가 손상되므로 m개의 비디오 패킷이 손실된다.)

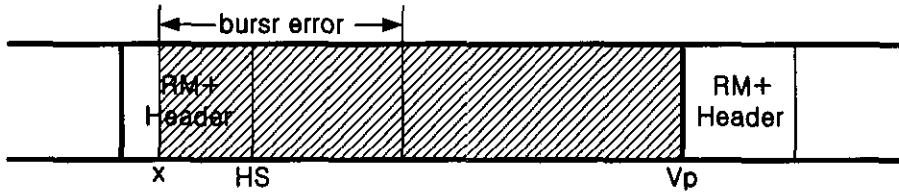


그림 8. 헤더에서 버스트 에러가 시작되는 경우

### 3.2.1 두 번째 비디오 패킷부터 버스트 에러의 영향.

$$loss_{2nd}$$

우선 하나의 버스트에러가 두 개 이상의 비디오 패킷에 걸쳐서 나타났을 경우, 두 번째 비디오 패킷에 잃어버리는 패킷의 개수는 다음과 같다. 비디오 패킷의 헤더가 손상되면 그 비디오 패킷은 버려야 한다. 따라서, 어떠한 에러 강인적 도구를 사용하더라도, 다음 식은 공통적을 유효하다. 왜냐하면 두 번째부터는 헤더가 먼저 손상되므로 에러 강인적 도구의 사용에 상관없이 비디오 패킷 전체를 버려야 하기 때문이다.

$$\begin{aligned} loss_{2nd} &= \sum_{x=1}^{V_p} \frac{1}{V_p} \sum_{i=1}^{\infty} \left[ i \sum_{j=i}^{(i+1)V_p-x} (1-p_1)^{j-i} p_1 \right] \\ &= \frac{1}{V_p} \sum_{x=1}^{V_p} \left[ \sum_{i=0}^{\infty} i \left\{ (1-p_1)^{iV_p-x} - (1-p_1)^{(i+1)V_p-x} \right\} \right] \\ &= \frac{1}{V_p \cdot p_1} [\text{Video packets/burst\_error}] \end{aligned}$$

$V_p$ 는 하나의 비디오 패킷의 길이이고,  $x$ 는 버스트 에러가 시작되는 시점을 말하고,  $l_b$ 는 버스트에러의 길이이다. 모두 비트 단위이다. 식에서  $\frac{1}{V_p}$ 는  $x$ 에서 버스트 에러가 시작될 확률을 말한다.  $i$ 는 손실되는 비디오 패킷의 개수이며, 그 뒤는  $i$ 개의 비디오 패킷이 손실될 확률을 말한다. 이제, 버스트에러가 하나의 비디오 패킷에서 시작하고, 그 비디오 패킷에서 끝난 경우만 생각하면 된다.

### 3.2.2 비디오 패킷 헤더에서 버스트에러가 시작되는 경우, $loss_{Header}$

즉,  $1 \leq x \leq HeaderSize$ 이며, 그림 8 (a)의 경우를 말한다. 시작된 비디오 패킷에서 버스트 에러가 끝났다고 해도, 헤더가 손상되는 경우에는 에러 강인적 도구에 상관없이 그 영향은 같다.

$$\begin{aligned} loss_{Header} &= \frac{1}{V_p} \sum_{x=1}^{HeaderSize} \sum_{i=1}^{\infty} (1-p_1)^{i-1} p_1 \\ &= \frac{HeaderSize}{V_p} [\text{Video packets/burst\_error}] \end{aligned}$$

### 3.2.3 프레임 헤더 (VOP 헤더)에서 에러가 생기는 경우, $loss_{VOPHeader}$

해당 VOP내의 모든 비디오 패킷을 잃어버린다.

$$loss_{VOPHeader} = \frac{VOPHeaderSize}{VOPSize} \text{NumberOfVideoPacketsInAVOP}$$

### 3.3 RM만 사용하는 경우

하나의 비디오 프레임을 여러 개의 비디오 패킷으로 나누어 전송하면 에러가 전파되는 것을 막을 수 있다. 하나의 비디오 프레임이 하나의 비디오 패킷으로 구성된다면 그것은 RM을 사용하지 않은 것과 같은 효과이다. 3.2.1절과 3.2.2절에서 공통적으로 고려한 경우를 제외하면, 이제 데이터 필드에서 버스트 에러가 시작되어 시작한 비디오 패킷에서 끝나는 경우만 고려하면 된다. 즉,  $HeaderSize < x \leq V_p$ 인 경우이다.

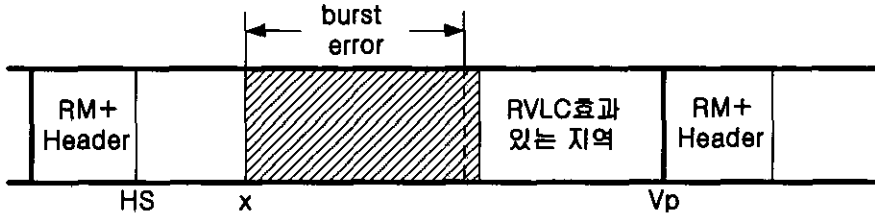


그림 9. RVLC가 효과가 있는 경우 (에러 지역의 마지막 매크로 블록이 끝날 때까지는 정보가 손실된다.)

$$loss_{RMData} = \frac{1}{V_p} \sum_{x=HS+1}^{V_p} \frac{V_p-x}{V_p-HS} \sum_{i=1}^{\infty} (1-p_1)^{i-1} p_1$$

여기서, HS는 HeaderSize이다. 일단, 에러가 시작되면 다음 비디오 패킷이 시작하기 전까지는 모두 버려야한다. 위 식을 정리하면,

$$loss_{RMData} = \frac{1}{V_p} \sum_{x=HS+1}^{V_p} \frac{V_p-x}{V_p-HS}$$

$V_p-x$ 를  $y$ 로 치환하여 정리하면,

$$\begin{aligned} &= \frac{1}{V_p(V_p-HeaderSize)} \sum_{y=0}^{V_p-HeaderSize-1} y \\ &= \frac{V_p-HeaderSize-1}{2V_p} \end{aligned}$$

위 식은 하나의 버스트 에러가 비디오 패킷 내부에서 발생하여 그 비디오 패킷내부에서 끝났을 경우 잃어버리는 정보의 양을 비디오 패킷 단위로 표시한 것이다. 따라서, RM을 사용하였을 경우, 하나의 버스트 에러에 의해서 잃어버리는 비디오 패킷의 개수는 다음과 같다.

$$loss_{RM} = loss_{common} + loss_{RMdata}$$

### 3.4 RM과 RVLC를 사용하는 경우

버스트 에러가 두 개 이상의 비디오 패킷에 걸쳐 있으면 RVLC의 효과는 전혀 없다. 그러나, 버스트 에러가 하나의 비디오 패킷에 국한되면, 뒷 부분으로부터 다시 디코딩이 가능하므로 뒷부분의 일부를 살릴 수 있다. 3.2.1절과 3.2.2절에서 공통적으로 고려한 경우를 제외하면, 이제 데이터 필드에서 버스트 에러가 시작되어 시작한 비디오 패킷에서 끝나는 경우만 고려하면 된다. 즉,  $HeaderSize < x \leq V_p$ 인 경우이다.

$$loss_{RVLCData} = \frac{\sum_{x=HS+1}^{V_p} \left( \sum_{i=1}^x (1-p_1)^{i-1} p_1 + \sum_{i=x+1}^{\infty} (V_p-x)(1-p_1)^{i-1} p_1 \right)}{V_p(V_p-HS)}$$

정리하면,

$$loss_{RVLCData} = \frac{1}{V_p(V_p-HS)} \sum_{x=HS+1}^{V_p} \frac{1-(1-p_1)^{V_p-x}}{p_1}$$

$V_p-x$ 를  $y$ 로 치환하여 정리하면,

$$\begin{aligned} loss_{RVLCData} &= \frac{1}{V_p(V_p-HS)p_1} \sum_{y=0}^{V_p-HS-1} [1-(1-p_1)^y] \\ &= \frac{1}{V_p p_1} - \frac{1-(1-p_1)^{V_p-HS}}{V_p(V_p-HS)p_1^2} \end{aligned}$$

따라서, RM과 RVLC를 동시에 사용하였을 경우, 하나의 버스트 에러에 의해서 잃어버리는 비디오 패킷의 개수는 다음과 같다.

$$loss_{RVLC} = loss_{common} + loss_{RVLCdata}$$

### 3.5 RM과 DP를 사용하는 경우

하나의 비디오 패킷이 움직임 벡터 (MV) 지역, 모션 마커 (MM), DCT 지역 (DCT)로 나누어 진다. 모션 마커는 MM에서 에러가 발생하고, 그 지역 내에서 끝난 경우, DCT 지역이 다시 시작하는 시점을 찾기위한 재동기 마커이다. DP를 사용하면, 매크로 블록은 다음 3가지 경우로 나누어 진다.

- a. MV, DCT 모두 있는 매크로 블록
- b. MV는 있으나, DCT는 없는 매크로 블록
- c. MV, DCT 모두 없는 매크로 블록

DCT는 있으나, MV가 없는 경우는 모두 없는 것과 마찬가지이다.

C만 손실로 간주하는 경우와 B와 C를 모두 손실로 간주하는 두 경우로 나눌 수 있다. 각각, DPc와 DPbc로 정의하자.

#### 3.5.1 MV 지역에서 버스트 에러 시작

MV지역내에서 끝난 경우에는 DCT 데이터는 모두 무시한 것으로 간주되나, MV지역을 지나 에러가 계속되면, DCT는 모두 손실된다. 우선 DPbc의 경우는 다음과 같다.

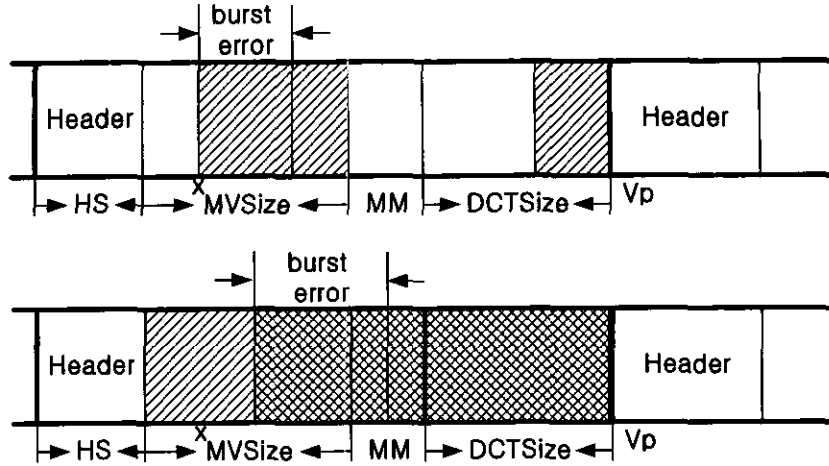


그림 10. (a) 버스트 에러가 MV 지역에서 시작하여 MV 지역에서 끝난 경우, (b) 버스트 에러가 MV 지역에서 시작해서 MV 지역을 넘어서 끝난 경우

$$\begin{aligned}
 loss_{DPbcMV} &= \frac{1}{V_p} \sum_{x=HS+1}^{HSMV} \left( \frac{HSMV-x}{MVSize} \sum_{l=1}^{HSMV-x} (1-p_1)^{l-1} p_1 \right. \\
 &\quad \left. + \sum_{l=HSMV+1-x}^{\infty} (1-p_1)^{l-1} p_1 \right) \\
 &= \frac{MVSize-1+2/p_1}{2V_p} - \frac{(1-p_1)(1-(1-p_1)^{MVSize})}{V_p p_1^2 MVSize}
 \end{aligned}$$

여기서 MVSize는 한 비디오 패킷에서 MV 데이터의 총 길이를 비트단위로 나타낸 것이며, HSMV = HeaderSize + MVSize이다. DPc인 경우에는 위에서 첫째항은 없어도 된다. 왜냐하면 DCT가 모두 무사하기 때문이다. 두 번째 항에서도, MV만 있는 매크로블록도 잃어버리지 않은 것으로 간주하므로,

$$\begin{aligned}
 loss_{DPcMV} &= \frac{1}{V_p} \sum_{x=HS+1}^{HSMV} \left( \frac{HSMV-x}{MVSize} \sum_{l=HSMV+1-x}^{\infty} (1-p_1)^{l-1} p_1 \right) \\
 &= \frac{1}{V_p} \sum_{x=HS+1}^{HSMV} \left( \frac{HSMV-x}{MVSize} (1-p_1)^{HSMV-x} \right) \\
 &= \frac{1}{V_p MVSize} \sum_{y=0}^{MVSize-1} y (1-p_1)^y \\
 &= \frac{1}{p_1 V_p MVSize} \left( -(MVSize-1)(1-p_1)^{MVSize} + \frac{1-p_1-(1-p_1)^{MVSize}}{p_1} \right)
 \end{aligned}$$

### 3.5.2 MM 지역에서 버스트 에러 시작

이 경우에는 MV 데이터는 모두 안전하므로, DPc의 경우에는 손실이 없어서  $loss_{DPcMV}=0$ 이다. 그러나, DCT 데이터를 모두 잃어버린다. 그러므로 DPbc인 경우에는 비디오 패킷이 모두 손실된 것

로 간주되므로,

$$loss_{DPbcMM} = \frac{1}{V_p} \sum_{x=HSMV+1}^{HSMV+MM} \sum_{l=1}^{\infty} (1-p_1)^{l-1} p_1 = \frac{MM}{V_p}$$

### 3.5.3 DCT 지역에서 버스트 에러 시작

이 경우에는 MV 데이터는 모두 안전하므로, DPc의 경우에는 손실이 없어서  $loss_{DPcMV}=0$ 이다. 그러나, DCT 데이터를 일부 잃어버린다. 그러므로 DPbc인 경우에는 비디오 패킷이 일부 손실된 것으로 간주되어서,

$$\begin{aligned}
 loss_{DPbcDCT} &= \frac{1}{V_p} \sum_{x=HSMV+MM+1}^{V_p} \frac{V_p-x}{DCTSize} \sum_{l=1}^{\infty} (1-p_1)^{l-1} p_1 \\
 &= \frac{1}{V_p} \sum_{x=HSMV+MM+1}^{V_p} \frac{V_p-x}{DCTSize} \\
 &= \frac{1}{V_p DCTSize} \sum_{y=0}^{DCTSize-1} y = \frac{DCTSize-1}{2V_p}
 \end{aligned}$$

여기서,  $DCTSize = V_p - HS - MV - MM$ 이다.

## IV. 실험 및 결과

일반적인 CDMA 환경에서 한 예를 들어보면 다음과 같다. MPEG-4의 에러강인성 코덱을 만들 때, 에러율은 다음과 같이 놓고 실험하였다<sup>[12]</sup>.

$$average\ BER = p_a = 1e-3$$

$$BAD\ State\ BER = p_b = 0.5$$

$$GOOD\ BER = p_g = 0.0$$

랜덤 에러는 데이터 링크 계층의 FEC (forward error correction)으로 제거되는 것으로 가정하고, 버스트 에러만 고려한다. MPEG-4 에러 강인성 연구 그룹에서는 버스트 에러의 길이를 1ms와 10ms으로 놓고 실험할 것을 권고하였다<sup>[3]</sup>. 본 연구에서는 0.1 ms에서 160 ms 등 40개의 길이에 대하여 실험하였다. 384kbps일 때, 버스트 에러의 길이는 0.1 ms일 때, 38비트 (약 5바이트)이고, 160 ms일 때 61440 비트 (7680바이트)이다.

비트율 r은 144kbps와 384kbps로 실험하였고, [9]에서 제안한대로 이중 75%를 비디오 전송에 사용하는 것을 가정하였다. 비디오 프레임율 fps는 144kbps에서는 1초에 10 프레임, 384kbps에서는 1초에 30 프레임으로 하였다. MPEG-4 비디오 패킷의 구조에서와 같이 HeaderSize는 50 bits로 하였고, motion marker MM은 17비트로 하였다. MVRatio는 비디오의 종류와 원하는 비트율에 따라 상당히 가변적인 값을 가진다. 드라마와 같이 비교적 움직임이 적은 비디오의 경우 384 kbps에서 MVRatio는 0.1 정도이다. 움직임이 많을수록, 높은 Q값을 사용할수록 MVRatio는 커져서, 0.5정도인 경우도 있다.

영상의 크기는 CIF 즉, 352×288로 하였다. 따라서, 하나의 프레임에서 매크로 블록의 개수 MBs는 396개이다. 한 비디오 프레임에서 비디오 패킷의 개수 VPs는 3GPP에서는 4-5개 정도로 권고하였고<sup>[9]</sup>, 본 실험에서는 1, 5, 20개로 실험하였다. 따라서, 하나의 비디오 패킷당 매크로 블록의 개수는 각각 평균 396, 80, 20개 정도이다. 따라서, 384 kbps, 30 Hz일 때 한 프레임의 평균 길이는 12,800비트이고, 비디오 패킷의 평균길이는 5개로 나눌 때 2560비트, 20개로 나눌 때 640비트이다.

3절에서 제시한 분석 방법을 실제와 비교하기 위하여 마이크로 소프트사 [11]가 제공하는 MPEG-4 인코더/디코더 소프트웨어를 이용하였다. 에러의 패턴은 DoCoMo [10]에서 제공하는 소프트웨어를 이용하여 발생하였다.

#### 4.1 에러강인성 도구 효과의 한계

BAD State에 속한 정보는 복원할 수 없다. 에러 강인성 도구의 역할은 BAD State의 영향이 GOOD State로 전파되는 정도를 최소화하는 것이다. 위와 같은 조건에서 전체에서 BAD State가 차지하는 비율은  $p_a = \frac{\bar{l}_b p_b + \bar{l}_g p_g}{\bar{l}_g + \bar{l}_b}$ 에서 구할 수 있다. 이 식에

서  $p_g=0$ 을 대입하면 식 (2.1)과 같이

$$\frac{\bar{l}_b}{\bar{l}_g + \bar{l}_b} = \frac{p_a}{p_b} \text{ 이고, 본 실험에서는 } 2 \times 10^{-3} \text{이다.}$$

이것이 에러 강인성 도구를 사용하여 다다를 수 있는 한계, LossLimit이다. 어떤 두 가지 서로 다른 방법을 비교할 때, 두 가지의 손실률의 차이가 10% 이내일 때 차이가 없는 것으로 간주하였다.

#### 4.2 RM의 효과

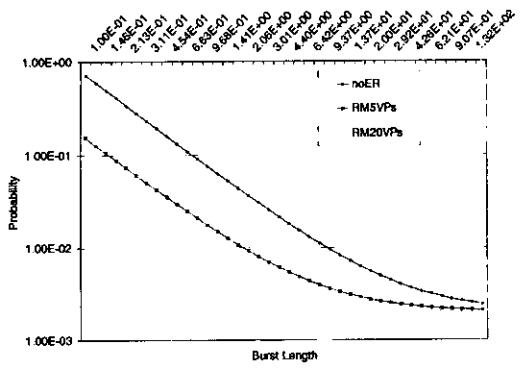
하나의 비디오 프레임은 더 많은 비디오 패킷으로 나눌수록 RM의 효과는 커진다. RM의 효과는 프레임을 비디오 패킷으로 나누지 않은 것과 비교해보면 알 수 있다. 그림에서 보면 RM을 사용하지 않았을 경우 (noER), 버스트 에러의 평균 길이  $\bar{l}_b$ 가 한 프레임 길이를 넘어서면 LossLimit에 도달한다. 즉, 한 프레임의 길이가 100ms인 경우인 10Hz에서는 약 132ms에서 LossLimit에 도달하였고, 한 프레임의 길이가 33ms인 경우에는 약 40ms에서 LossLimit에 도달하였다. 즉, 하나의 비디오 패킷의 크기가 비디오 프레임의 크기와 비슷해지면 RM의 효과는 없어진다. 또한 버스트 에러의 길이가 비디오 프레임의 길이보다 커지면 RM의 효과는 없다. 프레임율이 초당 10 (30) 프레임일 때, 한 프레임은 100 (33) ms이고, 따라서 버스트 에러의 크기가 100 (40) ms 정도되면 RM의 효과는 없다. 왜냐하면, 각 프레임은 이미 프레임 헤더로 재동기가 될 수 있게 되어있으므로, RM을 이용하여 재동기 (resynchronization)할 필요가 없기 때문이다.

더 많은 비디오 패킷으로 나눌수록 더 빨리 LossLimit에 도달한다. 즉, 144kbps, 10Hz인 경우에 한 프레임을 5개의 비디오 패킷으로 하는 경우 (RM5VPs)에는  $\bar{l}_b$ 가 24ms일 때 LossLimit에 도달하고, 20개로 나눌 때는 6.4ms이다. 이 길이는 비디오 패킷 길이보다 약간 큰 값이다. 즉, 비디오 패킷의 길이는 버스트 에러의 평균 길이보다 약 20% 작게 정하는 것이 최적이다. 384kbps, 30Hz에서는 5개로 나누는 경우와 20개로 나누는 경우 비디오 패킷의 길이는 각각 약 6.6ms와 1.6ms인데, LossLimit에 도달하는  $\bar{l}_b$ 는 각각 7.8ms와 2.05ms이다.

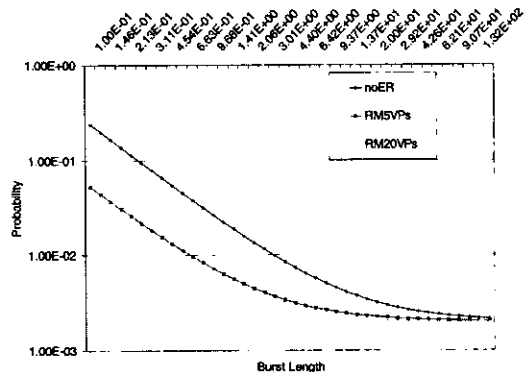
#### 4.3 RVLC의 효과

RVLC의 효과는 비디오 패킷 길이가 같을 때, RVLC를 사용한 경우와 사용하지 않은 경우를 비교하면 알 수 있다. 실험을 통해보면 RVLC의 한계





(a) 144kbps, 10Hz



(b) 384kbps, 30Hz

그림 11. RM의 효과

는 버스트 에러의 평균 길이가 비디오 패킷의 길이와 같아질 때이다. 왜냐하면, RVLC는 하나의 비디오 패킷에서 중간에 에러가 있을 경우, 양쪽에서 디코딩을 할 수 있어서, 효과를 보는 것이므로 버스트 에러의 크기가 비디오 패킷 길이보다 길어지면 효과가 없게 된다. 또한, 버스트 에러의 길이가 하나의 비디오 패킷 보다 커지면, 두 번째 비디오 패킷 부터는 모두 헤더가 손상되므로 비디오 패킷 전체를 버려야 하기 때문이다.

144kbps, 10Hz일 때, RM을 이용하지 않으면 (RVLC1VP),  $\bar{T}_b$ 가 약 75ms일 때 LossLimit에 도달한다. 비디오 패킷 각각 5개 ( $V_p=20ms$ )와 20개 ( $V_p=5ms$ )일 때에는  $\bar{T}_b$ 가 각각 16.5ms (RVLC5VPs)와 4.4ms (RVLC20VPs)일 때, LossLimit에 도달한다. 비슷한 시점에서 RVLC를 사용하지 않은 경우도 (RM5VPs, RM20VPs) LossLimit에 도달하므로,  $\bar{T}_b$ 가 비디오 패킷의 길이와 비슷하면 RVLC를 사용할 필요가 없다는 결론

이 나온다.  $\bar{T}_b$ 가 비디오 패킷 길이의 1/10 정도일 때, RVLC를 사용하여 에러율을 50%정도 줄일 수 있다. 즉, RVLC는 구현하기는 매우 복잡하지만, 버스트 에러 환경에서는 큰 효과가 없다.

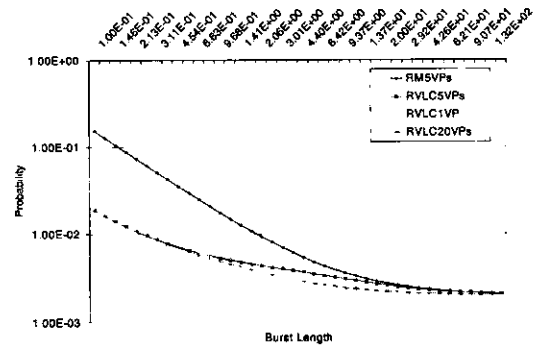


그림 12. RVLC의 효과

#### 4.4 DP의 효과

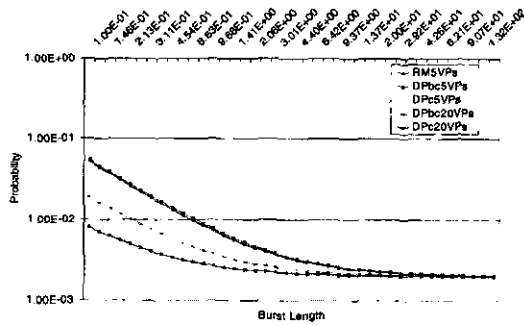
DP의 효과는 움직임 벡터 (MV)만 있는 매크로 블록을 손실로 했을 때 (DPbc)와 손실로 하지 않았을 때 (DPc)의 차이로 알 수 있다. DP의 한계도 버스트 에러의 평균 길이가 비디오 패킷의 길이와 같아질 때이다. DP도 RVLC와 마찬가지로 버스트 에러가 시작되는 첫 번째 비디오 패킷에만 효과가 있기 때문이다. DP의 효과는, 3.5절에서 언급한 것과 같이, 움직임 벡터는 있고 DCT는 없는 매크로 블록 (DPbc)이 얼마나 있는가로 판단된다. 왜냐하면, DP를 사용하지 않았으면, 그 매크로블록들의 움직임 벡터까지 대부분 잃어버렸을 것이기 때문이다.

적어도 움직임 벡터는 있는 매크로 블록 (DPc)에 대한 손실율은 RM만 있을 때보다 훨씬 빨리 LossLimit에 도달한다. 또한, MVRatio가 작을수록 움직임 벡터에 손실이 생길 가능성이 적어지므로 더 빨리 LossLimit에 도달한다.

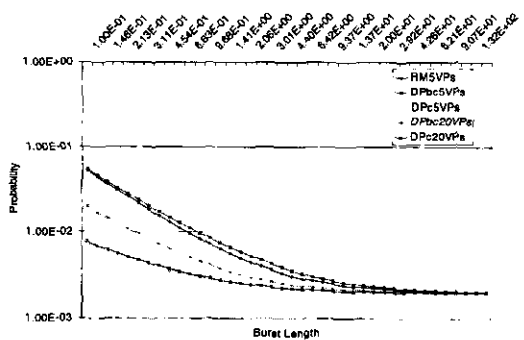
144kbps, 10Hz일 때, RM을 이용하지 않으면,  $\bar{T}_b$ 가 약 132ms일 때 DPc1VP와 DPbc1VP가 10% 이내로 근접한다. 비디오 패킷 각각 5개 ( $V_p=20ms$ )와 20개 ( $V_p=5ms$ )일 때에는  $\bar{T}_b$ 가 각각 29ms (DPbc5VPs, DPc5VPs)와 6.4ms (DPbc20VPs, DPc20VPs)일 때, 10% 이내로 근접한다. 즉,  $\bar{T}_b$ 가 비디오 패킷보다 30%이상 길 때에는 DP를 사용하는 효과가 없다는 결론이 나온다.

384kbps, 30Hz일 때, RM을 이용하지 않으면,

$\bar{T}_b$ 가 약 50ms일 때 DPc1VP와 DPbc1VP가 10% 이내로 근접한다. 비디오 패킷 각각 5개 ( $V_p=6.6ms$ )와 20개 ( $V_p=1.6ms$ )일 때에는  $\bar{T}_b$ 가 각각 9.4ms (DPbc5VPs, DPc5VPs)와 2.1ms (DPbc20VPs, DPc20VPs)일 때, 10% 이내로 근접한다. 즉, 144kbps, 10Hz의 경우와 마찬가지로  $\bar{T}_b$ 가 비디오 패킷보다 30% 이상 길 때에는 DP를 사용하는 효과가 없다는 결론이 나온다.



(a) MVRatio = 0.1



(b) MVRatio = 0.5

그림 13. DP의 효과

DP의 효과는 움직임 벡터가 전체 데이터에서 차지하는 비율에 따라서도 달라진다. 지금까지의 결과는 MVRatio가 0.1일 때의 결과이고, 0.5가 되면 DP가 효과있는 범위가 더 넓어진다. 144kbps, 10Hz일 때, RM을 이용하지 않으면,  $\bar{T}_b$ 가 약 200ms일 때 DPc1VP와 DPbc1VP가 10% 이내로 근접한다. 비디오 패킷 각각 5개 ( $V_p=20ms$ )와 20개 ( $V_p=5ms$ )일 때에는  $\bar{T}_b$ 가 각각 51ms (DPbc5VPs, DPc5VPs)와 11.3ms (DPbc20VPs, DPc20VPs)일 때, 10% 이내로 근접한다. 즉,  $\bar{T}_b$ 가

비디오 패킷보다 두 배 이상 길 때에도 DP가 효과가 있다는 결론이 나온다.

#### 4.5 결과

위 실험의 결과, MPEG-4 에러 강인성 도구의 효과는 버스트 에러의 평균 길이  $\bar{T}_b$ 에 크게 좌우된다는 것을 알 수 있다. 우선 RVLC는  $\bar{T}_b$ 가 비디오 패킷의 길이 이하일 때 효과가 있으며, DP도  $\bar{T}_b$ 가 비디오 패킷 길이보다 약간 큰 정도일 경우에 효과가 있지 그 보다 더 길어지면 사용하는 효과가 없다는 것을 알 수 있다.

비디오 패킷의 길이는  $\bar{T}_b$ 보다 20%정도 작은 값으로 정하는 것이 효과적이며, 이 때에는 DP를 이용하여 효과를 볼 수 있다. 버스트 에러의 평균 길이  $\bar{T}_b$ 가 비디오 한 프레임의 길이보다 길어지면, RM의 효과는 없어진다. 다음 그림은 각각의 에러 강인성 도구가 효과적인 범위를  $\bar{T}_b$  값에 대해서 나타내었다.

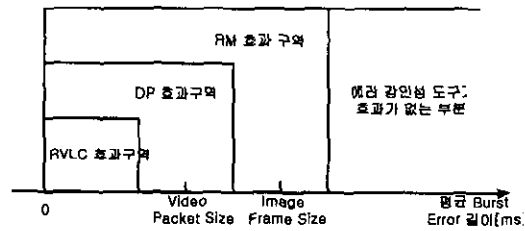


그림 14. 버스트 에러의 평균 길이에 대한 에러 강인성 도구의 효과

### V. 결론

IMT-2000의 비디오 서비스에서 버스트한 에러에 대한 대책으로 MPEG-4 비디오에서 제공하는 오류 강인성 도구들을 사용할 경우 그 효과에 대해서 분석했다. 그리고 버스트에러의 길이에 따른 오류강인성 도구들의 효율성을 분석했다.

오류강인성 도구들의 효과를 알아보는 방법으로 확실적인 분석방법을 사용했으며, 컴퓨터 시뮬레이션의 결과와의 비교를 통해서 그 타당성을 확인할 수 있었다. 컴퓨터 시뮬레이션을 통한 분석 방법은 많은 노력을 요구할 뿐만 아니라 발생 확률이 작은 경우에 대해서는 신뢰성을 확보하기 어렵다. 이런 단점을 보완해 줄 수 있는 확실적인 분석 방법은 매우 유용한 수단이라 할 수 있다.

오류강인성 도구 중의 하나인 재동기 마커 (RM)의 경우, 버스트에러의 길이가 프레임 길이보다 작을 경우는 효과가 좋으나, 버스트에러의 길이가 커지면 효과가 작아진다. 프레임율이 30Hz이면, 한 프레임의 길이가 33ms이므로, 버스트 에러의 길이가 10ms정도인 환경에서는 RM을 사용하여 효과를 볼 수 있다. 가역가변코드 (RVLC)와 데이터 분할 (DP)는 버스트에러의 크기가 비디오 패킷보다 작을 경우만 효과가 있다. 또한 다른 오류강인성 도구인 데이터 분할은 그 효과를 움직임벡터가 손실되지 않은 정도로 나타낼 수 있다. 이 경우 버스트에러의 길이가 비디오 패킷보다 크면 그 효과는 거의 없으며, 비디오 패킷보다 작을 경우는 전체 데이터에서 움직임 벡터의 비율이 작을수록 효과가 좋다.

손실이 발생하는 경우에 손실의 위치에 따라, 그 영향이 많이 차이가 나는 것을 알 수 있다. 따라서, 최근 IETF(Internet Engineering Task Force)에서는 중요한 정도에 따라, 비디오 계층의 하위 계층 (트랜스포트 계층)에서 손실에 대한 대책을 다르게 적용하는 ULP (unequal loss protection)을 위한 표준화가 진행되고 있다. 이 경우 I 프레임, 헤더 등에는 손실율이 다르게 된다. 본 논문에서는 이러한 경우는 다루지 않고 있으나, 이러한 시도는 계층화 압축 (scalable encoding)과 결합하여 이용하면 매우 효과적일 것이며, 본 논문에서 사용된 방법론을 확대 적용하여 분석할 수 있을 것이다.

참 고 문 헌

[1] Takehiro Moriya, Takeshi Mori, Naoki Iwakami, and Akio Jin, "A design of error robust scalable coder based on MPEG-4/Audio", Proceedings of the IEEE 2000 International Symposium on Circuits and Systems, Vol. III, pp213-216, May 2000.

[2] M.Podolsky, C.Romer, S.McCanne, "Simulation of FEC-based error control for packet audio on the internet," INFORCOM, San Francisco, VOL. II, pp.505-515, CA, March 1998.

[3] MPEG-4 video, "Information technology -coding of audio-visual objects: Part 2" ISO/IEC JTC1/SC29/WG11 N2502, FDIS 14496-2.

[4] MPEG-2, "Information technology-Generic

coding of moving pictures and associated audio information : Video" ISO/IEC JTC1/SC29/WG11, Final Text of 13818-2.

[5] Doug Young Suh, Hyun Cheol Kim, Youg Kwon Lim, Myung Ho Lee, "Probabilistic analysis of MPEG-4 error resilience tools in W-CDMA environments", Proceedings of the IEEE 2000 International Symposium on Circuits and Systems, Vol. III, pp638-641, May 2000.

[6] E. K. Wesel, Wireless Multimedia Communications, Addison Wesley, 1998.

[7] K. Otani, K. Kazuhiro and H. Omori, "Burst Error Performance Encountered in Digital Land Mobile Radio Channel," IEEE Trans. Veh. Techno., vol. VT-30, pp 156-160, Nov. 1981.

[8] B. Girod, K. Stuh., M. Link and U. Horn, "Packet Loss Resilient Internet Video Streaming," in part of IS&T/SPIE Conference on Visual Com. and Image Pro. Jan. 1999.

[9] 3GPP TSG-SA, "Quantitative performance evaluation of H.324 Annex C over 3G", TR26.116 v0.0.2, Apr. 1999.

[10] Toshio Miki and Toshiro Kawahara, "Error Patterns for Error Resilience Core Experiments," ISO/IEC JTC/SC29/WG11, MPEG96/1492, Nov. 1996.

[11] Microsoft MPEG-4 Encoder/decoder Software.

[12] MPEG-4 Requirements.

서 덕 영(Doug Young Suh)



1980년 2월 : 서울대 핵공학과 (공학사)

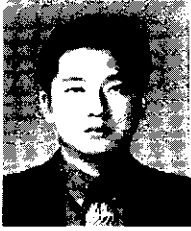
1986년 2월 : 미국 Georgia Tech. 핵공학 (공학석사)

1990년 6월 : 미국 Georgia Tech. 전기공학과 (공학박사)

1990년~1992년 : 상공부 생산기술 연구원, HDTV 연구개발단 선임연구원

1992년 3월~현재 : 경희대학교 전자정보학부 교수 <주관심 분야> 멀티미디어 통신, 인터넷, ATM.

홍 인 기( Ee n Kee Hong)



1985년~1989월: 연세대  
전기공학과 학사  
1989년~1991월: 연세대  
전기공학과 공학석사  
1991년~1995월: 연세대  
전기공학과 공학박사

1995년~1999년: SK Telecom 중앙연구원 선임연구  
원

1997년~1998년: NTT DoCoMo 교환 연구원

1999년~현재: 경희대학교 전자정보학부 조교수

<주관심 분야> CDMA, 이동통신

유 인 태( In Tea Ryoo)



1987년: 연세대학교 전자공학과  
학사  
1989년: 연세대학교 대학원  
전자공학과 석사  
1994년: 연세대학교 대학원  
전자공학과 박사  
1997년: 동경대학 대학원  
정보통신전공 박사

1997년~1999년: (주)삼성전자 네트워크 사업부 선임  
연구원

1999년~현재: 경희대학교 전자정보학부 조교수

<주관심 분야> 컴퓨터 네트워크, 초고속 정보 통신 이  
론, 차세대 인터넷 프로토콜, 네트워크 QoS,  
멀티미디어 통신 시스템