

# MPEG-4에서 오류 강인성을 위한 오류전파 제한방법에 대한 연구

\*이 상 조,\*서 덕 영,\*\*임영권,\*\*이명호

\*경희대학교 전자 정보학부,\*\*한국전자통신 연구원

## A Study On Error Localization Techniques for MPEG-4 Error Resilience

\*Sang-Jo Lee,\*Doug Young Suh,\*\*Young-Kwon Lim,\*\*Myoung-Ho Lee

\*Dept. of Electronic Eng., Kyung-Hee University

\*\*ETRI

sjlee@earth.kyunghee.ac.kr

### 요약

MPEG-4에서 오류강인성(Error Resilience)를 위한 한 방법으로 Resynchronization Markers(RM)을 사용한다. 한 프레임이 시작될 때 StartCode를 사용하여 동기를 맞추고 몇 개의 MacroBlock을 encoding한 후 일정한 비트수(Threshold 값)가 지나면 재동기 마커 표시하여 재동기를 한다. 이렇게 하므로써 한 프레임 내에서 어떤 부분에 에러가 발생하더라도 그 에러가 속해있는 비디오패킷(재동기 마커와 재동기 마커사이의 Data)만을 버리거나 RVLC(Reversible Variable Length Codes)를 사용하여 Data를 복원할 수 있다. 그러나 만약 재동기 마커에 에러가 발생하거나 에러의 전파로 인하여 재동기 마커를 인식 못한다면 두 개 이상의 패킷이 손실되거나 RVLC를 사용한 데이터 복원을 할 수 없다.

본 논문에서는 이를 막기위해 디코딩 전에 Prescan을 통해서 재동기 마커의 위치를 탐지하고 에러가 생긴 재동기 마커를 복원하는 방법을 제안하였다. 그리고 bitrate에 따른 MB(Macro Block)의 크기와 비디오 패킷 크기(재동기 마커와 재동기 마커간의 거리)를 분석하여 재동기 마커를 찾는 루틴에 적용하였다.

### 1.서론

MPEG-4에서 오류강인성(Error resilience)을 위한 방법으로 주기적으로 재동기시키는 resynchronization과 RVLC를 이용한 data recovery 그리고 Data Partition을 이용한 에러 은닉(Error concealment)이 있다.

먼저, Resynchronization(재동기)는 VOP내의 비트스트림에 주기적(일정하지는 않음)으로 Resynchronization Marker(RM)을 표시하므로써 오류가 존재하는 한 패킷(재동기 마커와 재동기 마커간의 비트스트림)만을 버려서 오류가 확산되는 것을 막을 수 있다.

그리고, RVLC는 순방향뿐만 아니라 역방향으로도 디코딩 될 수 있도록 제안된 코드로서 두 재동기 마커사이에 오류가 발생되더라도

그 패킷 전체를 버리는 것이 아니라 뒤 오류가 검출된 패킷 다음에서부터 다시 디코딩을 해서 Data를 복구할 수 있도록 한다. 그리고, Data Partition은 스트림길이가 작지만 중요한 정보를 가지고 있는 motion과 비교적 긴 길이의 texture를 분리시키므로써 texture에서 에러가 발생하여 그 값들을 버리더라도 전 VOP의 texture값을 가져와서 현재 VOP에서 오류가 없는 것처럼(오류 은닉) 디코딩을 할 수 있다.

본 논문<sup>1)</sup>에서는 MPEG-4에서 오류강인성을 위한 위의 방법 중 Resynchronization Marker 부분에 오류가 생길 경우 이를 효과적으로 복원하여 패킷 손실을 줄일 수 있는 방법을 제

1) 본 연구는 ETRI 영상 처리 연구팀 수탁과제의 일부임.

안한다.

본 논문의 구성은 다음과 같다. 제 2장에서 MPEG-4에서 재동기 마커의 중요성과 제안된 방법에 대해 설명하고, 제 3장에서는 재동기 마커가 나타나는 위치를 분석해서 한 비디오패킷의 크기 분포를 알아보고, 제 4 장에서는 디코딩 전 Prescanning 을 통한 재동기 마커의 위치 파악 및 에러 복원을 하고 그 효율성에 대해 분석한다. 마지막으로 제 5장에서는 제안된 방법에 대한 결론으로 끝을 맺는다.

## 2. 재동기 마커의 중요성과 제안

MPEG-4에서 오류강인성을 위한 3가지 방법을 적용하는데 있어서 재동기 마커는 반드시 사용되고 RVLC나 Data Partition방법을 사용하는데 있어서 재동기 마커를 정확하게 인식하고 있어야 위 방법의 적용이 가능하다. 또한 다른 Data와는 달리 vop start code나 재동기 마커처럼 동기를 맞추어주는 코드에 에러가 난다면 다음 재동기까지의 Data를 모두 잃어버리는 결과를 초래한다.

본 논문에서는 MPEG-4로 인코딩되어서 전송된 Data를 디코딩하기 전에 Prescan을 통해서 재동기 마커의 위치를 알아내고 에러가 발생한 재동기 마커를 복원하여 오류강인성을 위한 방법을 적용하는데 있어 효과를 높이고자 한다.

재동기 마커는 00000000 00000000 1의 17비트의 특정한 코드가 들어가는데 이중 1비트에 러나 2비트에러가 발생하였을 경우까지 재동기 마커으로 인식하여 실제로 어느정도의 복원 효율이 있고 어떤 방법을 사용하는 것이 더 효과적인지를 살펴보겠다.

다음장에서는 에러가 발생한 재동기 마커를 찾기 위해 재동기 마커가 어느정도의 간격으로 존재하는지 분석해 보겠다.

## 3. 재동기 마커의 위치 분석

먼저 MPEG-4에서 사용되는 VOP의 구조 먼저 살펴보겠다.

VOP를 인코딩하면서 SC에서 먼저 동기를 맞

추고 MB단위로 인코딩을 하면서 비트를 세는데 이 값이 정해진 Threshold값 이상이 되면 현재 인코딩하던 MB까지 인코딩을 마치고 Byte align을 하기 위해 패딩을 한 후에 재동기 마커를 삽입한다. 이런식으로 한 VOP가 끝날 때까지 재동기 마커를 삽입하면서 인코딩을 하게 된다.

### 3.1 통계적으로 보았을 때 재동기 마커와 재동기 마커간(패킷)의 길이 분석

MPEG-4 에서의 비디오 패킷의 길이는 Threshold값과 연관이 있다. 비디오 패킷의 길이 분포는 Threshold값부터 나타나기 시작해서 이 값 이후 인코딩되던 매크로 블록이 끝날 때까지 사용된 비트수까지는 계속 나타난다. 패킷크기의 확률 분포를 알아보는데 사용된 파일은 Coast영상을 Threshold값 각각 1000, 2000, 3000비트로 주고 384kbps로 인코딩한 파일을 사용하였다.

각각의 크기분포도를 살펴보면 모두 동일한 특성의 분포를 갖는다는 것을 알 수 있다.

패킷크기의 확률분포는

$$P = \frac{1}{\lambda} e^{-\frac{1}{\lambda}(t - t_0)}$$

의 식으로 표현할 수 있을 것이다.

여기서  $\lambda$  = 평균 패킷의 크기 - Threshold값  
 $t_0$  = Threshold 값이다. 여기서 위에서 사용된 세 파일의  $\lambda$  평균은 80이 나왔다.

아래의 그림은 실제 패킷의 크기를 분석한 확률 분포와 위의 확률 분포 함수를 같이 표현하였다.

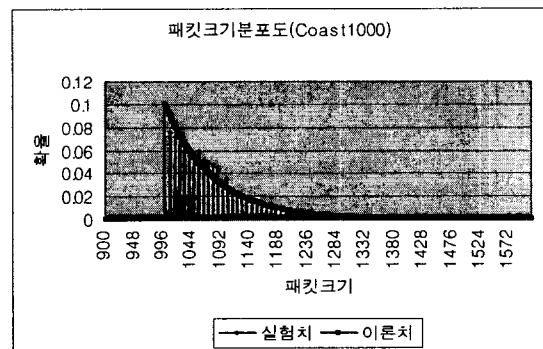


그림1. 패킷크기분포(Coast1000)

패킷의 크기 분포는 Threshold값을 넘는 그 비디오 패킷의 마지막 MB의 크기와 연관이 있는데 다음 절에는 이 MB의 크기분포를 살펴 보겠다.

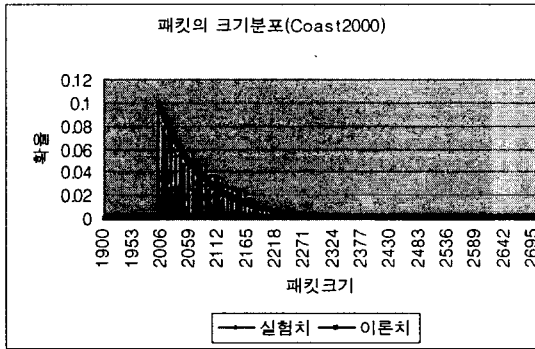


그림2. 패킷크기분포(Coast2000)

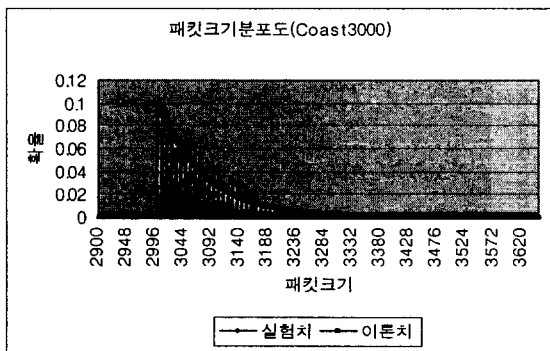


그림3. 패킷크기분포(Coast3000)

### 3.2 매크로 블록의 크기 분석

위에서 사용된 각각의 Coast 영상은 384kbps로 인코딩된 100frames의 cif파일이다. 이 파일의 총 크기는 472kBytes이므로 평균 MB의 크기 S를 이론적으로 알아보면

$$S = \frac{472000 * 8 \text{비트}}{(100 \text{frame} * 396 \text{MB})} = 95 \text{비트/MB} \text{ 이다.}$$

다음 그림은 각 파일별로 MB의 크기를 분석한 것인데 거의 비슷한 분포를 갖는다. 이 MB의 평균 크기는 95.5 bit로서 이론적으로 계산한 결과 값과 거의 같게 나왔다. 전 절에서 통계적으로 구했던  $\lambda$  값 80과 MB의 평균 크기 95는 깊은 연관이 있다.

다음 장에서는 Threshold값 이후 나타나는 재동기 마커만을 인식하도록한 재동기 마커에러복원 알고리즘을 이용하여 재동기 마커를

찾아보겠다.

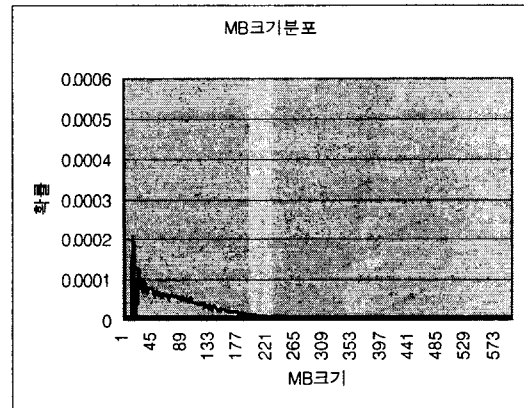


그림4. 패킷크기분포(Coast1000)

## 4. Pre-Scanning을 통한 재동기 마커의 위치 파악 및 에러복원

MPEG-4로 인코딩되어서 전송된 자료를 디코딩하기 전에 Pre-Scan을 통해 먼저 재동기 마커의 위치를 알아내고 에러가 발생한 재동기 마커를 복원하는 알고리즘과 그 효율을 알아보겠다.

재동기 마커의 위치 파악 및 에러복원에 사용된 파일은 384k비트율로 인코딩하고 Threshold값이 각각 1000비트, 2000비트, 3000비트인 300 frames의 Coast파일을 대상으로 분석해 보았다. 에러율은 E-3, E-4, E-5, E-6, E-7의 5가지로 적용하였고 에러복원 루틴은 재동기 마커에서 1비트 또는 2비트 에러가 발생할 수 있다고 가정하여 재동기 마커의 에러를 복원하고 찾았다.

### 4.1 재동기 마커에서 1bit, 2bitError 찾는 알고리즘

그림7 순서도에서 DetectOnebitSCError()의 알고리즘은 Buffer에서 읽은 3Bytes값과 StartCode 00000000 00000000 00000001을 비교하여 1비트 차이까지는 허용하여 SC로 인식한다. DetectTwobitSCError()의 알고리즘은 SC와 2비트 차이까지 허용하여 SC로 인식한다.

DetectOnebitRMError()의 알고리즘은 재동기 마커 00000000 00000000 1과 Buffer에서 읽은

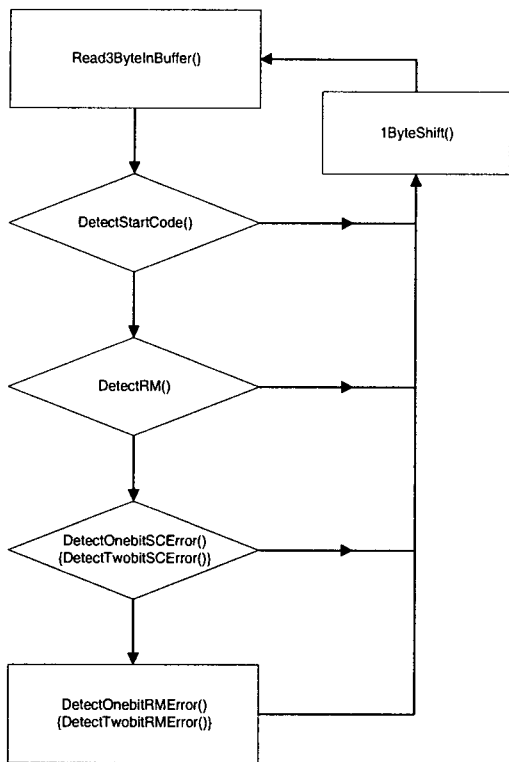


그림 5. 재동기 마커에러 Detect 순서도

17비트값을 비교하여 1비트 차이까지는 재동기 마커으로 인식한다. 단, 전의 SC나 재동기 마커보다 Threshold값 이상 떨어진 부분에 존재하는 것만을 재동기 마커으로 인식한다.

DetectTwobitRMErrror()의 알고리즘도 재동기 마커와 2비트 차이까지 재동기 마커으로 인식하는 것외에는 DetectOnebitRMErrror()의 알고리즘과 같다.

#### 4.2 에러율에 따른 재동기 마커에 에러가 발생될 확률 분석

재동기 마커는 0000 0000 0000 0000 1의 17비트로 표시하고 이 17비트에서 에러가 발생될 확률  $P=1-(1-BER)^{17}$  이다. 여기서 BER은 bit에 에러가 발생될 확률이다. 그러므로 n개의 재동기 마커중에서 에러가 발생한 재동기 마커의 개수는  $N_{ERROR\_RM} = N_{RM} * \{1-(1-BER)^{17}\}$  이다.

다음 표는 에러율에 따른 에러가 발생한 재동기 마커의 수를 이론치와 실험치를 비교하였다. 이론치와 실험치와는 약간의 차이가

있는데 이는 비교적 짧은 Data인 300 frames만으로 분석한 결과에 따른 것이다.

		Coast 1000	Coast 2000	Coast 3000	
재동기 마커갯수		3623	1894	1301	
에러가 발생한 재동기 마커의 갯수	E-3	이론치	61.10	31.94	21.94
		실험치	55	23	26
	E-4	이론치	6.15	3.22	2.21
		실험치	3	2	2
	E-5	이론치	0.62	0.32	0.22
		실험치	0	1	0
	E-6	이론치	0.06	0.03	0.02
		실험치	0	0	0
	E-7	이론치	0.01	0.00	0.00
		실험치	0	0	0

표.1.에러율에 따른 에러가 발생한 재동기 마커의 개수

다음은 재동기 마커에 1비트 에러(해밍 디스턴스 1)와 2비트 에러(해밍 디스턴스 2)가 발생될 확률에 대해 알아보겠다. 재동기 마커에서 n개의 에러가 발생될 확률  $P_n=17C_n*BER^n*(1-BER)^{17-n}$ 이므로

\* 재동기 마커의 해밍 디스턴스가 1일 확률  $P_1=17C_1*BER*(1-BER)^{16}$  이고,

\* 재동기 마커의 해밍 디스턴스가 2일 확률  $P_2=17C_2*BER^2*(1-BER)^{15}$  이다.

만약 BER이 E-3이라면  $P_1=1.673*E-2$ 이고,  $P_2=1.675*E-5$ 이다. 여기서 재동기 마커의 해밍 디스턴스가 2일 확률은 해밍 디스턴스가 1일 확률에 BER배만큼 훨씬 작음을 알 수 있다.

본 논문에서는 에러가 발생한 재동기 마커를 복원하는데 해밍 디스턴스가 1인 경우와 2인 경우 2가지를 가정하여 재동기 마커를 찾아보았는데 다음 섹션에서는 2가지 방법 중 어떤 것이 더 효율적인지를 알아보도록 하겠다.

### 4.3 재동기 마커의 해밍 디스턴스가 1일 경우 복원 효율

재동기 마커에 1비트의 에러가 발생할 수 있다고 가정하고 재동기 마커를 찾아 보았다. BER이 E-4 ~ E-7 일때는 에러가 발생한 재동기 마커를 다 찾을 수 있었고 BER이 E-3 일 겨우는 에러가 발생한 재동기 마커를 90% 이상 찾을 수 있었다. 또한 재동기 마커가 아닌 Data를 재동기 마커으로 인식하는 경우도 없었다.

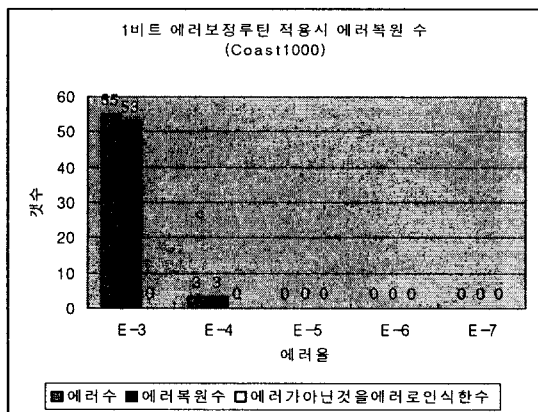


그림6. 1비트 에러보정시 에러복원 수(T:1000)

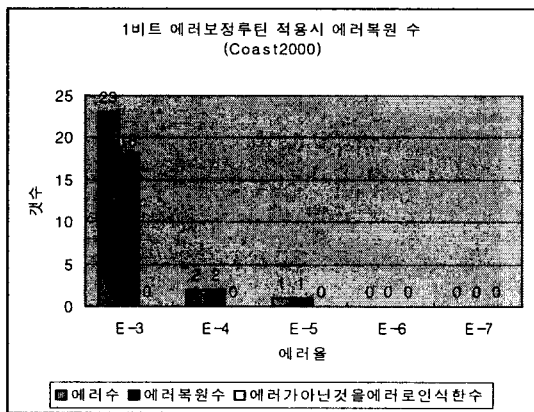


그림7. 1비트 에러보정시 에러복원 수(T:2000)

### 4.4 재동기 마커의 해밍 디스턴스가 2일 경우 복원 효율

재동기 마커에 2비트의 에러가 발생할 수 있다고 가정하고 재동기 마커를 찾아보았는데 BER이 E-4~E-7까지의 에러가 발생한 재동

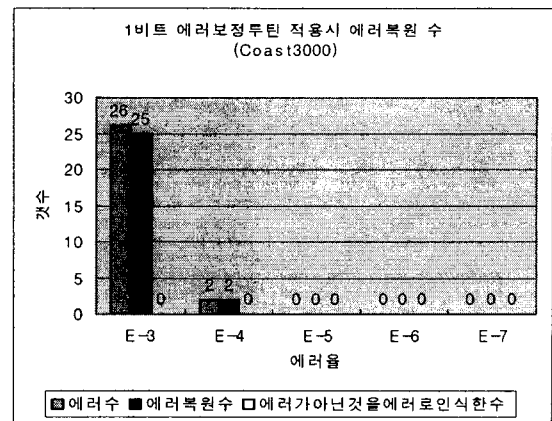


그림 8. 1비트 에러보정시 에러복원 수(T:3000)

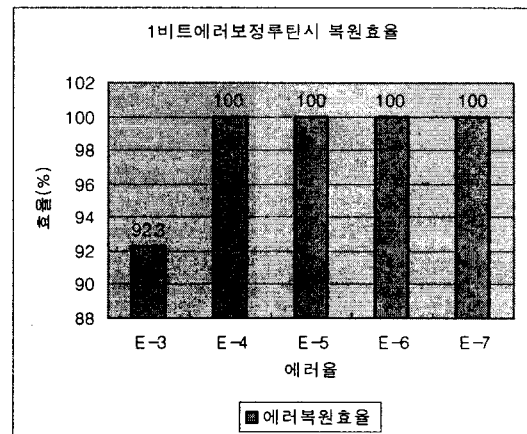


그림 9. 1비트 에러복원 효율

기 마커는 다 찾을 수 있었다. 그러나 BER이 E-3일 경우 에러가 발생한 재동기 마커를 90% 이상 찾았지만 재동기 마커가 아닌 Data를 재동기 마커으로 인식한 경우도 있었다. 종합적으로 보면 재동기 마커의 에러복원 효율은 90.3%이다.

이상에서 살펴보았듯이 1비트 에러 보정루틴이나 2비트 에러 보정루틴을 적용하여 재동기 마커를 찾는다면 BER이 E-4~E-7 범위에서는 두가지방법 모두 100% 재동기 마커의 오류를 찾는 뛰어난 효과가 있었고 비교적 BER이 큰 E-3일 경우에는 1비트 에러 보정루틴의 효율이 92.3%로 2비트 에러 보정루틴 90.3%보다 낫다는 것을 알 수 있다.

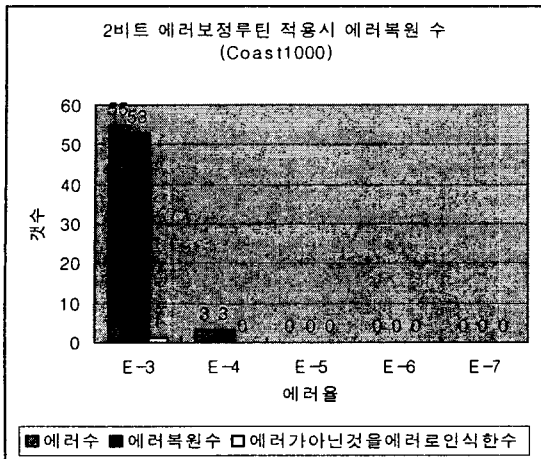


그림 10. 2비트 에러보정시 에러복원 수(T:1000)

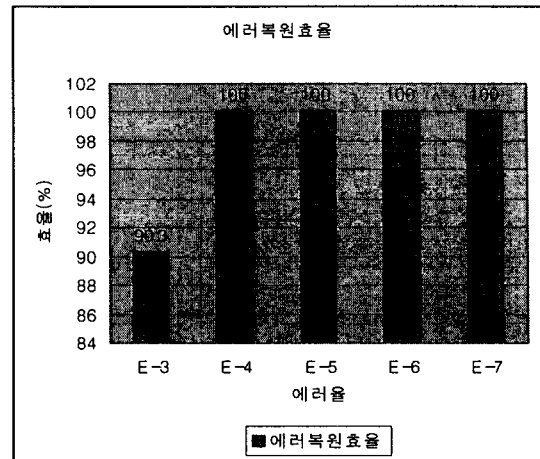


그림 13. 2비트 에러복원 효율

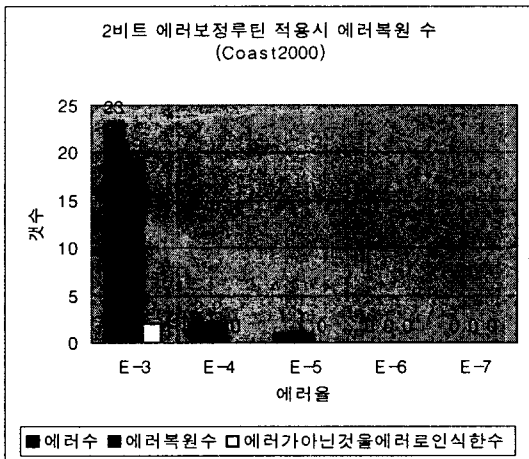


그림 11. 2비트 에러보정시 에러복원 수(T:2000)

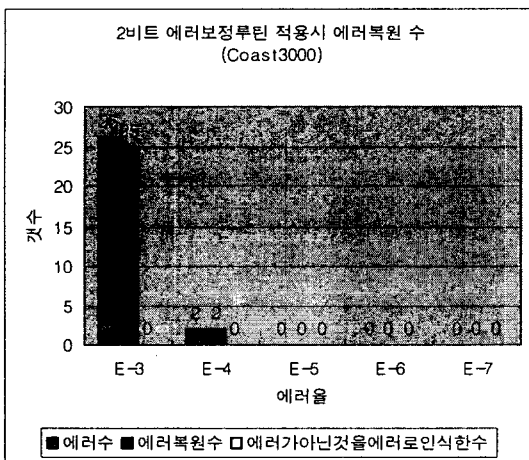


그림 12. 2비트 에러보정시 에러복원 수(T:3000)

또한 2비트 에러 보정루틴의 경우에는 재동기 마커가 아닌 17비트의 값을 재동기 마커로 인식하는 경우도 있었다.

### 5. 결론

디코딩전에 Pre-Scanning을 통해서 재동기 마커의 위치를 탐지하고 에러가 생긴 재동기 마커를 복원하는 알고리즘을 사용한다면 패킷 내의 에러전파로 인한 재동기 마커를 인식 못하는 경우를 없애고 1비트 에러 보정루틴을 사용한 오류정정 알고리즘으로 재동기 마커의 오류로 인한 패킷 전체의 Loss를 줄일 수 있으며 오류 강인성을 위한 방법들을 적용하는데 있어서 더 효과적인 결과를 얻을 수 있을 것이다.

### 참고문헌

1. 김현철, "IMT-2000에서 MPEG-4 비디오 오류 강인성 연구", 제12회 신호처리 합동 학술대회 논문, pp703-706, 1999.
2. 석주명, "QoS를 고려한 인터넷 기반의 MPEG-4 비디오 폰 및 VOD 구현", 1999년 경희대학교 졸업 논문.
3. "MPEG-4 VM 12.1", ISO/IEC JTC/ SC29 /WG11 N2552
4. PEYTON Z. PEEBLES, JR, "PROBABILITY, RANDOM VARIABLES, AND RANDOM SIGNAL PRINCIPLES", Third Edition, 1991.