

# Realtime 전송을 위해 RTP를 사용한 Error Correction Scheme 연구

박 덕 근 , 박 원 배  
경북대학교 정보통신학과  
전화 (053)940-8896 / (053)940-8559

## A study of error correction scheme using RTP for real-time transmission

Durk Gun Park , Won Bae Park  
Dept. of Information Communication, Kyungpook National University  
E-mail: [papi17@knuicsvr2.knu.ac.kr](mailto:papi17@knuicsvr2.knu.ac.kr) / [wbpark@ee.knu.ac.kr](mailto:wbpark@ee.knu.ac.kr)

### Abstract

A forward error correction (FEC) is usually used to correct the errors of the real-time data occurred at the receiver side which require a real-time transmission. The data transmission is performed after being encapsulating by RTP and UDP. In the ITU-T study group 16, four FEC schemes using the XORing are presented.

In the paper, a new supplementary scheme is proposed.

In the delay problem the new scheme performs better than the scheme 3 but in the recovery ability for successive packet loss is worse than scheme 3. The proposed scheme which supplements the present schemes can be adapted easily to the current network environment

실시간 처리를 요하는 데이터의 인터넷상에서 전송시 에러가 발생하거나 패킷이 분실된 경우 많은 재전송 시간이 필요한 ARQ(Automatic Repeat Request)를 사용할 수 없으므로 FEC방법을 많이 사용하고 있다.

FEC는 원래의 신호인 미디어데이터가 분실하더라도 수신측에서 분실되지 않은 나머지 패킷을 이용하여 분실된 패킷을 복원시키는 방법이다.

FEC는 큰 지연없이 손실 패킷을 복구할 수 있는 장점을 가지고 있으나 패킷의 에러나 손실 특성에 따라 적절한 변화가 필요하다.

I 서 론

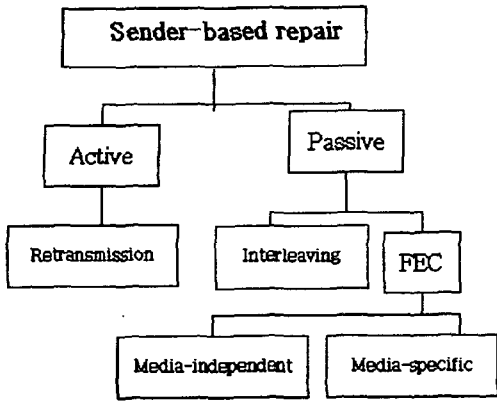


Figure 1. Sender-based repair

이에 따라 ITU-T의 Study Group 16에서는 Media-independent Error Correction 방법으로 4가지의 XOR을 사용한 FEC scheme을 제안하였다.

Scheme은 real-time이라는 환경에 유리하도록 bandwidth와 latency 그리고 cost를 최소화하려고 했으며 이에 따라 네트워크환경의 변화에 따라 적절한 scheme을 적용시키기 위해 네 가지 scheme을 정하였다.

이 scheme에 의해 만들어진 error-correction을 위한 FEC데이터와 원래의 데이터는 RTP(Real-time Transfer Protocol)에 실리게 된다.

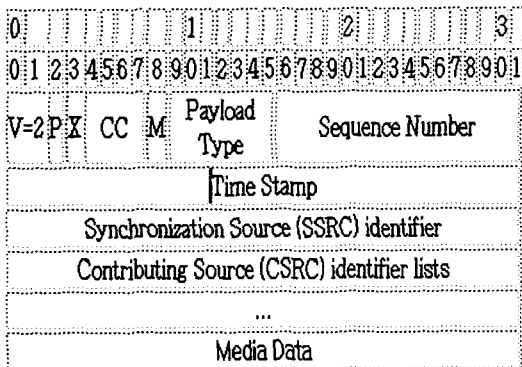


Figure 2. RTP Packet

실시간 데이터 전송에 적합한 전송 프로토콜인 RTP는 주로 오디오, 비디오 등의 실시간 데이터를 멀티캐스트나 유니캐스트로 전송하는데 사용된다. 일반적으로 응용계층수준에서 구현되는 RTP는 payload

를 식별하기 위한 확장된 구조와 함께 Sequence numbering과 time stamping을 제공함으로써 멀티미디어 데이터의 전송을 돕도록 설계되었다.

이렇게 RTP에 실린 데이터는 ARQ를 필요로하지 않는 UDP에 의해 다시 encapsulation되어 UDP의 다중화 및 체크섬 기능을 이용한다.

ITU-T에서 제안한 네 가지의 Scheme은 오버헤드와 지연시간의 크기가 차별화를 두어 네트워크 환경의 변화에 적용하도록 하였다. 그러나 변화가 심한 네트워크 환경에 보다 더 탄력적이며 효율적으로 적용하기 위해서 또 하나의 Scheme을 제안한다.

다섯 번째 Scheme은 네 번째 Scheme의 경우보다 연속적인 패킷의 손실에 대한 복원률이 떨어지지만 지연시간이 크지 않으며 Scheme 1과 2가 하지 못하는 4개의 연속적인 패킷분실에도 어느 정도 복원이 가능하다는 장점이 있다.

II장에서는 ITU-T에서 제안한 scheme과 비교하여 scheme 4의 특징과 효율을 알아보고 III장에서는 결론과 향후 남은 과제에 대해 논할 것이다.

## II 본론

FEC Scheme에 따른 특징을 (n:k:d)로 나타내는데 n은 원시 패킷수, k는 FEC 전송 패킷수, d는 패킷지연을 나타낸다.

### 1. Error-Correction Scheme 0 (1:1:0)

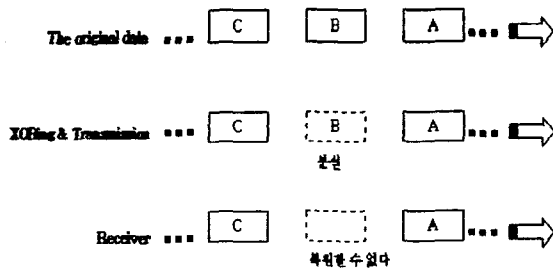


Figure 3. Scheme 0

Scheme 0은 별도의 FEC패킷을 생성하지 않고 그대로 전송하는 방법이다. 따라서 패킷 A,B,C를 보냈을 때 패

킷 B를 분실하였을 경우 수신측에서는 네트워크상에서 분실된 패킷을 복구할 정보를 가질 수 없다. 네트워크에서 손실율이 아주 낮을 경우에 사용하며 당연한 결과로 지연문제와 오버헤드문제가 없다

2. Error-Correction Scheme1 (2:1:1)

Scheme 1은 a,b,c,d...=> a,ab;b,bc;c,cd;처럼 두 개의 미디어 패킷 사이에 XOR한 FEC를 첨가해서 보낸다. 두 개의 패킷은 FEC패킷을 포함하여 3개의 패킷으로 전송된다. 따라서 1단위만큼의 delay가 생기게 된다.

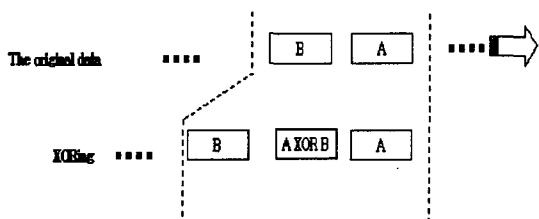


Figure 4. Scheme 1

3. Error-Correction Scheme2 (3:2:1)

Scheme2는 a,b,c,d,e,f,g => ab,ac,abc;cd,ce, cde;ef,eg,efg 형태로 XORing되어 전송된다.

여기서 복원시 carry-over의 문제가 생기며 따라서 특정 미디어 패킷이 복원되는나 아니냐는 다음 패킷의 복원에 영향을 미친다. 여기서 Carry-over는 그룹으로 나뉘어져 전송할 때 한 그룹의 제일 끝의 미디어 패킷이 된다. 복원하는 방법은 다음과 같다.

Carry-over 값이 있을 경우

$$b = (a) (ab) = (ac) (abc)$$

$$c = (a) (ac) = (ab) (abc)$$

Carry-over 값이 없을 경우

$$a = (ab) (ac) (abc)$$

$$b = (ac) (abc)$$

$$c = (ab) (abc)$$

1개또는 2개가 연속으로 분실되었을 경우에도 복원이 가능하나 원래의 미디어 패킷만큼의 오버헤드가 생긴다.

Scheme 2의 경우 연속이 아닌 부분적으로 분실된 모든 패킷을 복원할 수 있으며 2개의 연속적인 패킷 손실에 도 70~80% 정도의 복구율을 보인다. 이러한 복구율이

Scheme1에 비해 다소 떨어지더라도 오버헤드를 낮출 수 있는 장점이 있다.

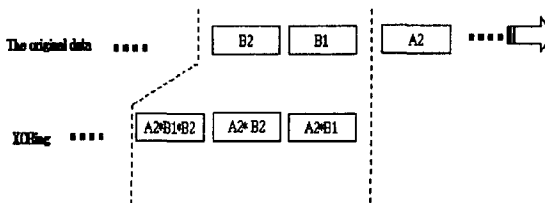


Figure 5. Scheme 2

4. Error-Correction Scheme3 (2:1:4)

Scheme3은 a,b,c,d => a,b,abc,c,acd,abd,d,bcd 형태로 전송되며 네 가지 Scheme중에 연속적인 패킷의 손실 복구에 가장 강점을 가진다. 이러한 장점의 결과로 지연시간 면에서는 좋지않은 결과를 가져 올 수밖에 없다.

즉, 4개의 데이터당 4개의 오버헤드를 가지게 되고 한 패킷 그룹당 4단위의 delaytime을 가지게 된다.

Scheme 3은 1,2,3개의 연속적인 패킷 분실에는 물론 4개 연속적인 패킷 분실 네트워크 환경에서도 약 80% 정도의 복구가 가능하다.

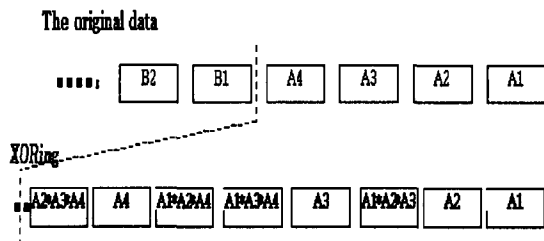


Figure 6. Scheme 3

복원방법은 아래와 같다.

$$a = (a) = (b) (c) (abc) = (b) (d) (abd)$$

$$= (c) (d) (acd) = (d) (abc) (bcd) = (c) (abd) (bcd)$$

$$= (b) (acd) (bcd) = (abc) (abd) (acd)$$

$$b = (b) = (a) (c) (abc) = (a) (d) (abd) = (c) (d) (bcd) = (d)$$

$$(abc) (acd) = (c) (abd) (acd)$$

$$= (a) (bcd) (acd) = (abc) (abd) (bcd)$$

5. Error-Correction Scheme4 (2:1:3)

새롭게 제시하는 scheme이며 scheme 2와 scheme 3의 중

간 역할을 효율적으로 할 수 있을 것으로 예상된다.

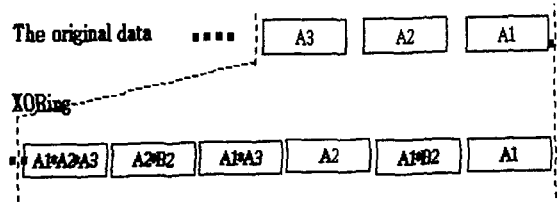


Figure 7. Scheme 4

3개의 패킷 그룹에 3개의 오버헤드를 가지게 되며 delaytime은 3만큼 발생하게 된다.

$a,b,c;d,e.. \Rightarrow a,ab,b,ac,bc,abc;d,de..$

이 새로운 Scheme은 패킷 그룹끼리는 독립적이어서 다른 그룹에 영향을 받지 않는다.

Scheme4의 복원 방법에는 5가지가 있다.

$a = (a) = (b) (ab) = (bc) (abc) = (b) (ac) (bc)$   
 $= (ab) (bc) (abc)$

$b = (b) = (a) (ab) = (ac) (abc) = (a) (ac) (bc)$   
 $= (ab) (bc) (abc)$

$c = (a) (ac) = (b) (bc) = (ab) (abc) = (a) (b) (abc)$   
 $= (ac) (bc) (abc)$

Scheme 3의 경우 각 패킷마다 복원 방법이 8가지나 되어 이로 인해 연속적인 패킷 분실에 강점을 가짐을 알 수 있다. 하지만 지연시간은 엄청나게 길어진다는 단점이 있는데 특히 Scheme3이 사용될 손실이 많은 네트워크 환경에서는 전송지연의 특징도 함께 나타나는 상황을 충분히 고려하여야 한다. 따라서 다섯 번째의 새로운 Scheme의 효용이 충분히 있음을 알 수 있다.

### III 결론

측정 결과 새로 고안한 이 다섯번째 scheme은 연속적으로 패킷을 분실한 경우 복원률은 60~70% 정도이다. scheme 3에 비해 작은 latency를 가지는 장점이 있는 반면 연속적으로 분실된 패킷에 대한 복원확률이 다소 떨어진다. 하지만 scheme 1과 2에 비해서는 연속적인 패킷 분실의 복원확률이 높아 네트워크환경에 따라 scheme 4를 사용하면 네 개의 scheme을 사용하여 분실패킷의 복원을

하는 경우보다 보다 효율적인 전송과 복원이 이루어질 것이다.

따라서 향후 여기서 제안한 scheme이 적용될 수 있는 가장 효율적인 네트워크환경을 분석하고 다른 scheme과의 적절한 전환이 이루어지는 포인트를 알아내는 연구가 필요하다.

### IV 참고 문헌

- [1].Bolot, J.-C. and Vega-Garcia,A., "The case for FEC-based error control for packet audio in the Internet," *Multimedia Systems*, 1997
- [2].H.Schulzrinne,S.Casner,R.Frederick,V.Jacobson, "RTP: A transport Protocol for Real-Time Applications," RFC 1889.
- [3].D.Budge,R.McKenzie,W.Mills,W.Diss,P.Long "Media-independent Error Correction using RTP" 1997
- [4].박준석,고대식 "인터넷폰의 설계 및 개발"
- [5].Colin Perkins,Orion Hodson,and Vicky Hardman "A Survey of Packet Loss Recovery Techniques for Streaming Audio"
- [6].Jean-Chrysostome Bolot, Sacha Fosse-Parisis, Don Towsley "Adaptive FEC-Based Error Control for Internet Telephony"
- [7].Jean-Chrysostome Bolot, Andres Vega-Garcia "Control Mechanisms for Packet Audio in the Internet"