

VoIP를 위한 적응적 FEC 기반 에러 제어 알고리즘

최 태 옥[†]·정 기 동^{**}

요 약

현재의 인터넷은 가변적인 대역폭과 패킷손실 그리고 지연으로 인하여 대화식 응용의 QoS 보장이 어렵다. 특히 최근에 정보의 기반구조로 중요성이 강조되고 있는 VoIP는 패킷손실률과 종점간지연이 클 때 통화품질이 크게 떨어지므로 네트워크 수준에서나 응용 수준에서 에러제어 기법이 요구된다. 인터넷 전화와 같은 대화식 응용을 위한 응용 수준의 에러 제어 기법으로 FEC(Forward Error Correction)가 가장 많이 사용되고 있는데, 이 기법은 주정보와 더불어 부가정보를 전송함으로써 패킷손실을 복구하는 방법으로 네트워크의 상태에 따라 적응적으로 부가정보의 양을 조절한다. 그러나 기존의 알고리즘들은 패킷손실만을 고려하여 부가정보를 조절하였으며 부가 정보를 증가시킬 때 수반되는 종점간지연을 간과함으로써 통화품질을 떨어뜨리는 단점이 있다. 본 논문에서는 패킷손실뿐만 아니라 종점간지연을 고려하는 FEC 기반 에러제어 기법인 SCCRP (Selecting a Codec Combination using Reward and Penalty)를 제안한다. 실험 결과, SCCRP는 다른 알고리즘들에 비해 복구 후 패킷손실률은 물론 복구 후 종점간지연을 낮게 유지하였다.

An Adaptive FEC based Error Control Algorithm for VoIP

Tae-Uk Choi[†] · Ki-Dong Chung^{**}

ABSTRACT

In the current Internet, the QoS of interactive applications is hardly guaranteed because of variable bandwidth, packet loss and delay. Moreover, VoIP which is becoming an important part of the information infra-structure in these days, is susceptible to network packet loss and end-to-end delay. Therefore, it needs error control mechanisms in network level or application level. The FEC-based error control mechanisms are used for interactive audio application such as VoIP. The FEC sends a main information along with redundant information to recover the lost packets and adjusts redundant information depending on network conditions to reduce the bandwidth overhead. However, because most of the error control mechanisms do not consider end-to-end delay but packet loss rate, their performances are poor. In this paper, we propose a new error control algorithm, SCCRP, considering packet loss rate as well as end-to-end delay. Through experiments, we confirm that the SCCRP has a lower packet loss rate and a lower end-to-end delay after reconstruction.

키워드 : 인터넷 전화(Internet Telephony), VoIP(Voice over IP), FEC(Forward Error Correction), 에러 제어(Error Control)

1. 서 론

최근 네트워크와 컴퓨터 기술의 발달로 인하여 인터넷상에서 음성 서비스를 지원할 수 있는 인터넷 전화(Internet Telephony)가 대두되고 있다. 인터넷 전화는 회선교환망 대신 인터넷망을 사용하기 때문에 전화요금을 현재 수준의 50%에서 80%까지 낮출 수 있고, 컴퓨터 기술과 결합되어 다양한 음성 애플리케이션 개발이 가능하므로, 국내외적으로 기존 통신사업자는 물론 인터넷 사업자들도 많은 관심을 두고 있다.

그러나, 현재의 인터넷 전화는 가변적인 대역폭과 손실, 지

연으로 일반전화만큼의 통화품질을 제공하지 못하고 있다. 일반적으로 인터넷에서 패킷손실률은 0%~20% 정도이고, 단방향 지연은 5ms~500ms 정도이다. 패킷손실률이 10%를 초과하거나 단방향 지연이 150ms 이상이 될 경우는 사용자가 느끼는 음질은 매우 저하된다. 이러한 문제를 해결하기 위해서 네트워크 차원에서 QoS를 보장하기 위한 연구들이 진행되고 있으며, 대표적으로 Differentiated Services[1]나 Integrated Services[2, 3]등이 있다. 현재의 다양한 네트워크는 향후 IP-기반 망으로 통합될 추세이며 best-effort service는 물론 integrated service, differentiated service 등의 다양한 서비스를 제공할 것이다.

그러나, 네트워크 차원에서 QoS 기능을 제공하더라도 단말 쪽의 응용 수준에서 QoS 조절이 요구된다. 즉, 네트워크가 부하 상태이거나 QoS를 완전히 보장하지 못할 경우 애플리

[†] 준 회원 : 부산대학교 대학원 전자계산학과

^{**} 종신회원 : 부산대학교 전자계산학과 교수

논문접수 : 2001년 11월 5일, 심사완료 : 2002년 2월 20일

케이션은 적용적으로 QoS를 조절할 수 있어야 한다. 응용 수준에서 QoS 제어는 패킷손실(loss), 지연(delay), 지터(jitter) 등의 네트워크 특성의 영향을 최소화시키는 방향으로 연구되고 있다. 지터에 의한 영향을 최소화시키기 위해 다양한 재생 버퍼(playout buffer) 알고리즘[4, 5]들이 연구되어 왔고, 패킷손실을 최소화하기 위한 방법으로 에러 제어 기법(error control scheme)이나 과잉 밀집 제어를 위한 전송율 제어 기법(rate control scheme)[6, 7] 등이 연구되었다.

패킷손실을 보상하기 위한 대표적인 기법들로는 interpolation이나 interleaving을 이용한 concealment 기법들[8, 9], 재전송(retransmission)에 기반한 에러 복구 기법들[10], FEC에 기반한 에러 제어 기법들이 있다[11-14]. Concealment 방법은 손실된 패킷자체를 복구하지 못하기 때문에 통화품질을 크게 향상시키지는 못한다. 그리고 재전송 방법은 수신측에서 손실된 패킷을 다시 전송하는 방법으로 재전송 지연 때문에 실시간 대화식 애플리케이션에는 적합하지 못하다. 주정보와 함께 부가정보를 전송하는 FEC 기법은 패킷이 손실될 경우 그것의 부가정보를 이용하여 원래의 패킷을 복구한다. 이 방법은 재전송 지연 없이 패킷 손실을 효과적으로 줄일 수 있기 때문에 인터넷 전화와 같은 대화식 응용에 가장 적합하다. 그러나 FEC는 모든 패킷에 부가 정보를 포함시키기 때문에 패킷손실률이 작을 때 대역폭이 낭비될 수 있다. 이를 극복하기 위한 방법으로 네트워크의 상태에 따라 동적으로 부가정보의 양을 조절하는 기법들이 연구되었다[13, 14]. 그러나 이들 기법들은 패킷손실률만을 고려하여 부가정보의 양을 조절하기 때문에 복구 후 패킷손실률은 감소시킬 수 있으나 반대급부로 전송률이 증가하고 종점간지연 시간이 증가한다는 점을 간과하였다. 인터넷 전화 응용에서는 종점간지연이 150ms 이상이 될 경우 음성 품질이 매우 저하되므로 부가정보의 양을 조절할 때 패킷손실률은 물론 종점간지연도 같이 고려해야 된다.

본 연구에서는 네트워크의 상태에 따라 중복정보의 양을 조절할 때 패킷 손실은 물론 종점간지연 시간을 고려하여 중복정보의 양을 결정할 수 있는 적응적 FEC 기반 에러 제어 기법인 SCCRP(Selecting a Codec Combination using Reward and Penalty)를 제안한다. 이 기법은 주정보와 부가정보를 각각 다른 코덱을 사용하여 인코딩하는 미디어 독립적 FEC 기법을 사용하며, 부가정보의 양을 조절하기 위해 코덱들의 조합을 교체한다.

실험을 위하여 실제 인터넷상에서 측정된 트레이스와 NS-2 시뮬레이터를 통해 수집된 트레이스들을 사용하여 제안된 기법의 성능을 측정하였다. 실험 결과, SCCRP는 복구 후 패킷손실률이나 복구 후 종점간지연에서 종점간지연을 고려하지 않는 다른 기법들에 비해 더 높은 성능을 유지함을 확

인하였다.

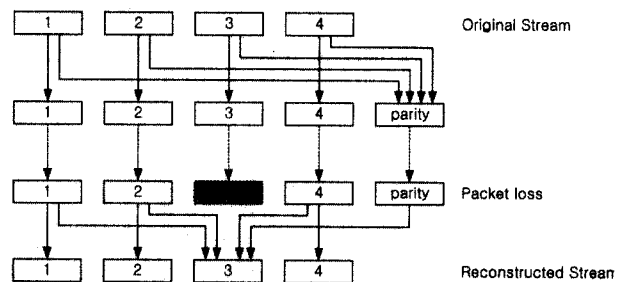
본문의 구성은 다음과 같다. 2장에서 FEC 기반 에러 제어 기법의 종류와 부가정보의 조절에 대해서 설명하고 3장에서는 reward와 penalty에 기반한 적응적 에러 제어 기법인 SCCRP를 설명한다. 4장에서 실제 인터넷 트레이스와 NS-2 트레이스를 이용하여 성능을 측정하고 5장에서 결론을 맺는다.

2. FEC 기반 에러 제어

2.1 미디어 독립적 FEC 와 미디어 종속적 FEC

대화식 인터넷 응용에서 가장 적합한 에러 제어 방법은 수신자 기반의 FEC 알고리즘이며 이것은 부가정보의 종류에 따라 미디어 독립적인(media independent) FEC와 미디어 종속적인(media specific) FEC로 나뉜다[15]. (그림 1)과 (그림 2)는 이 두 가지 FEC 기법을 보여준다.

미디어 독립적인 FEC는 수학적 연산을 이용하여 패리티 코드의 형태로 부가 정보를 생성한다. 이 기법은 미디어의 내용에 상관없이 적용할 수 있으며 대표적인 알고리즘으로 parity coding과 Reed-Solomon coding 알고리즘이 있다. Parity coding에서는 여러 개의 패킷 그룹에 대하여 XOR 연산을 수행함으로써 하나의 패리티 패킷을 생성한다. (그림 1)에서 볼 수 있듯이 만일 패리티를 생성하기 위해 사용된 패킷이 손실되었을 경우 패리티 코드를 통해서 복구될 수 있다. Rosenberg와 Schulzrinne은 다양한 형태의 패리티 코드를 구성하는 방법을 제안하였다[16].



(그림 1) 미디어 독립적 FEC

(그림 2) 미디어 종속적 FEC

미디어 종속적인 FEC는 샘플링된 하나의 오디오 유니트를 여러 개의 패킷에 중복 전송한다. 이 때 여러 개의 복사본중에서 처음 보내는 것을 primary encoding이라 하고 그 다음에 보내는 복사본들을 secondary encoding이라고 한다. 대부분 secondary encoding은 primary encoding보다 대역폭이 더 낮고, 질이 더 좋지 않고, 압축률이 높은 코덱을 이용한다. 이 기법은 Hardman과 Bolot 등에 의해 제안되었다[17, 18]. (그림 2)는 미디어 종속적인 FEC 기법의 예를 보여주는데 하나의 패킷 안의 두 개의 오디오 유니트를 포함한다. 원정보는 자신의 오디오 유니트를 primary encoding한 것이고 부가정보는 이전 오디오 유니트를 secondary encoding한 것이다. 만일 하나의 패킷이 손실된다면 다음 패킷의 부가정보에 포함되어 있는 secondary encoding을 이용하여 손실을 복구할 수 있다. 이 기법은 패리티 정보를 생성하고 복원하는 오버헤드가 없기 때문에 지연이 낮고 복구를 위해서 하나의 패킷만 기다리면 되기 때문에 종점간지연이 중요한 대화식 응용에 적합하다.

2.2 FEC 부가정보의 조절

FEC는 주정보에 부가정보를 추가해서 보내기 때문에 대역폭 낭비가 심하다. 이러한 대역폭 낭비를 최소화시키기 위하여 다양한 적응적 부가조절 기법들이 제안되었다. 즉, 네트워크의 상태가 원활하여 패킷 손실이 적을 때는 부가정보를 줄이고 네트워크 상태가 좋지 못할 경우는 부가정보의 양을 늘리는 것이다. 그러나 대부분의 오디오 코덱은 코딩률이 고정되어 있어 동적으로 코딩률을 조절하는 것이 쉽지 않다. 그래서 primary encoding 코덱이나 secondary encoding 코덱 자체를 대체시키는 기법이 이용된다. 이 때 primary codec과 secondary codec을 선택하는 것은 쉽지 않은 문제이며 시스템에서 지원 가능한 여러 가지 코덱 조합을 고려해야 한다. 즉, 미디어 종속적인 FEC에서는 전송률을 조절하기 위해서 사용가능한 코덱들의 조합을 구성하고 네트워크의 상태에 따라 적절한 코덱 조합을 선택해야 한다. 대표적인 기법으로 Bolot 알고리즘[13]과 USF 알고리즘[14]을 들 수 있다.

2.2.1 Bolot 알고리즘

Bolot 알고리즘[13]은 복구 후 손실률을 미리 정해진 LOW와 HIGH의 임계치 안으로 유지시킨다. 네트워크의 손실률이 HIGH 마크보다 높을 때는 부가정보를 증가시키고, 네트워크 손실률이 LOW 마크보다 낮을 때는 부가정보의 양을 줄인다. 이를 위해 이 알고리즘은 주정보와 부가정보를 생성하기 위한 코덱 조합들을 구성하고 이 조합들을 이용했을 때 얻을 수 있는 이득을 reward로 정량화하였다.

〈표 1〉 Bolot알고리즘의 코덱 조합들

No.	Codec Combination	Reward
0	(PCM)	1
1	(PCM, ADM4(1))	2.5
2	(PCM, GSM(1))	2.5
3	(PCM, LPC(1))	2.5
4	(PCM, ADM4(3))	6
5	(PCM, ADM4(1), ADM2(2))	6
6	(PCM, ADM4(1), ADM2(3))	10
7	(PCM, ADM4(1), ADM2(2), ADM2(3))	18

〈표 1〉은 주정보와 부가정보를 생성하기 위한 코덱 조합들과 reward값을 보여준다. 조합 No.1은 PCM으로 인코딩된 패킷 N의 주정보와 ADM4로 인코딩된 패킷 N-1의 부가정보를 의미한다. 코덱 조합 번호와 관련된 reward는 복구 전 패킷손실률을 복구 후 패킷손실률로 나눈 값으로 이전 패킷의 부가정보를 많이 보낼수록 복구 후 패킷손실률은 적어져 reward값은 커진다. 복구 후 패킷손실률과 복구 전 패킷손실률은 실제 여러 번의 실험을 통하여 측정된 평균 값이다.

Bolot 알고리즘은 RTCP 피드백 정보를 이용하여 현재의 패킷손실률을 구하고 reward 정보를 이용하여 복구 후 패킷손실률을 추정한다. 그리고 추정된 복구 후 패킷손실률이 HIGH 또는 LOW 임계치를 벗어났을 때 코덱 조합 번호를 증가하거나 감소함으로써 부가정보를 조절한다. 그러나 경험적으로 구해진 reward값이 모든 네트워크 환경을 대표하지는 않으며 이 값을 사용하여 구해진 복구 후 패킷손실률이 정확한 값이라고 보기 힘들다. 또한, 부가정보의 양을 조절했을 때 발생하는 네트워크 패킷손실률의 변화를 고려하지 않아서 코덱 조합 번호가 자주 증감을 반복하는 순환 현상(cyclical behavior)이 발생할 수 있다.

2.2.2 USF 알고리즘[20]

Bolot 알고리즘과 같이 USF 알고리즘에서도 네트워크의 패킷손실률에 기반하여 코덱 번호를 증감함으로써 부가정보를 조절한다. 그러나 reward값을 사용하여 복구 후 패킷손실률을 추정하지 않는다. 실제로 수신자는 복구되지 못한 패킷의 수를 측정하여 이 값을 RTCP 정보에 포함시켜 수신자에게 피드백한다. 수신자는 이를 바탕으로 복구 후 패킷손실률을 계산하고 이 값이 HIGH나 LOW 임계치를 벗어났는지를 비교한다. 또한, 이 알고리즘은 순환 현상을 방지하기 위해서 LOW 임계치 이상을 벗어났을 때 바로 코덱조합 번호를 감소시키지 않고 이전 복구 후 패킷손실률과 현재 복구 후 패킷손실률의 차이가 일정한 값(Minimal Threshold) 이상인 경우만 코덱 조합 번호를 감소시킨다.

그러나 이 방법은 수신자의 정보들(복구 전에 손실된 패킷의 개수, 복구 후에 손실된 패킷의 개수, 버스트한 손실의 개수)을 송신자에 보내기 위해서 RTCP 패킷을 확장해야 한다. 그리고 Bolot 알고리즘은 복구 후 패킷손실률이 높더라도 네트워크가 과잉 밀집될 경우에 전송률을 줄임으로써 패킷손실률을 줄이지만 이 방법은 네트워크가 과부하일 때를 고려하지 않고 있다.

3. Reward와 Penalty에 기반한 적응적 에러 제어 기법

3.1 Reward와 Penalty

적응적 FEC 기법에서 부가정보의 양을 많이 보낼수록 수신측의 패킷손실률을 낮출 수 있으나, 부가정보를 위한 계산 오버헤드, 버퍼 공간의 증가, 그리고 수신 측에서 부가정보를 포함하는 패킷을 기다리는 시간의 증가로 종점간지연 시간은 증가된다. 특히, 인터넷 전화에서 종점간지연은 통화품질에 크게 영향을 미친다. 따라서 패킷손실과 종점간지연은 서로 상반되는 관계이며 적절한 절충점을 선택해야 한다. 본 연구에서는 부가정보를 포함시킴으로써 얻을 수 있는 이득(패킷손실률)과 손실(종점간지연)을 reward와 penalty로 정량화 시킨다. 즉, 부가정보를 이용할 때와 이용하지 않을 때의 패킷손실률의 비율을 reward라 하고, 부가정보를 이용할 때와 이용하지 않을 때의 종점간지연 시간의 비율을 penalty라고 정의한다. 식 (1)과 식 (2)는 각각 reward와 penalty를 계산하기 위한 수식을 나타낸다.

$$reward = \frac{loss\ rate\ before\ reconstruction}{loss\ rate\ after\ reconstruction} \quad (1)$$

$$penalty = \frac{end-to-end\ delay\ after\ reconstruction}{end-to-end\ delay\ before\ reconstruction} \quad (2)$$

미디어 종속적인 FEC 기법은 주 정보를 코딩하기 위한 주 코덱(main codec)과 부가정보를 코딩하기 위한 부가 코덱(redundant codec)이 필요하며, 보통은 부가코덱의 데이터율은 주 코덱의 데이터율보다 낮다. 네트워크의 상태에 따라 부가정보를 조절하기 위해서는 FEC의 차수를 조절할 수도 있으며 차수가 동일할 경우는 부가 코덱의 종류를 달리할 수도 있다. 따라서 여러 개의 코덱이 있을 경우 주 코덱과 부가 코덱의 결합은 다양한 코덱 조합을 이룬다. <표 2>는 대표적인 주 코덱과 부가 코덱의 조합과 이를 사용해서 얻을 수 있는 reward와 penalty를 보여준다. 코덱 조합에서 처음에 오는 코덱이 주 코덱이고 뒤에 오는 코덱들이 부가 코덱들이다. 부가 코덱들이 포함하는 번호는 차수(order)라 부른다. 예를 들어 조합 번호 4에서 주정보는 G.711로 코딩되고 이전 패킷의 부가정보는 GSM으로 코딩되며 그 이전 패킷의 부가정보는 G.729

로 코딩됨을 의미한다. 각각의 코덱 조합에 해당하는 reward와 penalty는 실제로 인터넷에서 음성데이터를 전송하여 얻어지는 패킷손실과 종점간지연을 바탕으로 계산되었다. <표 2>에서 볼 수 있듯이 부가 정보가 증가함에 따라 reward와 penalty가 증가함을 알 수 있다.

<표 2> 코덱 조합에 따른 reward와 penalty 값

번호	코덱 조합	reward	penalty
0	(G.711)	1	1
1	(G.711, GSM(1))	2.5	1.5
2	(G.711, G.723(1))	2.5	2
3	(GSM, G.723(1))	2.5	4
4	(G.711, GSM(1), G.729(2))	6	2.4
5	(G.729, G.723(1), LPC10(2))	6	4.5
6	(G.711, GSM(1), G.729(2), LPC10(3))	18	3.4
7	(G.711, GSM(1), G.723(2), LPC10(3))	18	4.5

<표 3> 주요 변수들

변수	설명
$N_{lost\ packets}$	현재 RTCP 주기안에서 손실된 패킷 개수
$N_{expected\ packets}$	현재 RTCP 주기안에서 기대한 패킷 개수
$N_{recovered\ packets}$	현재 RTCP 주기안에서 복구된 패킷 개수
$T_{display}(j)$	현재 RTCP 주기안에서 j번째 패킷의 재생 시간
$T_{sampling}(j)$	현재 RTCP 주기안에서 j번째 패킷의 샘플링 시간
Lb'_i	RTCP 주기 t에서 코덱 조합 i의 복구 전 패킷손실률
La'_i	RTCP 주기 t에서 코덱 조합 i의 복구 후 패킷손실률
Db'_i	RTCP 주기 t에서 코덱 조합 i의 복구 전 종점간지연
Da'_i	RTCP 주기 t에서 코덱 조합 i의 복구 후 종점간지연
Le_i	코덱 조합 i를 사용했을 때 예상되는 패킷손실률
De_i	코덱 조합 i를 사용했을 때 예상되는 종점간지연
R_i	코덱 조합 i의 reward 값
P_i	코덱 조합 i의 penalty 값
α	평활 계수

3.2 SCCRP(Selecting a Codec Combination using Reward and Penalty)

<표 2>에서 제시된 FEC 부가정보의 reward와 penalty는 실험을 통하여 측정된 값으로 다른 네트워크 환경에서는 그 값들이 달라질 수 있다. SCCRP는 보다 정확한 값들을 추출하기 위해서 RTCP 패킷을 이용하여 주기적으로, 적응적으로 이 값들을 갱신한다. 이를 위해서 RTCP 패킷을 확장하여 수신측의 복구 후 패킷 손실, 복구 후 지연 정보들을 RTCP 패킷 안에 포함시켜 피드백 한다. 송신측에서 먼저 주기 t에 도착한 RTCP 정보를 이용하여 현재 코딩중인 코덱 조합

i 의 복구 전 패킷손실률 Lb_i^t , 복구 후 패킷손실률 La_i^t 을 구한다.

$$Lb_i^t = \frac{N_{lost\ packets}}{N_{expected\ packets}} \quad (3)$$

$$La_i^t = \frac{N_{lost\ packets} - N_{recovered\ packets}}{N_{expected\ packets}} \quad (4)$$

그리고, RTCP 주기 t 에서 현재 코딩중인 코덱 조합 i 의 복구 후 종점간지연을 측정하기 위해서 식 (5)와 식 (6)를 이용한다. n_i 가 주기 t 에서 전송된 패킷의 개수라고 할 때 주기 t 의 복구 후 종점간지연 Da_i^t 은 주기내 각 패킷의 종점간지연 시간의 평균이다. 또한, 복구 전 종점간지연 Db_i^t 은 복구 후 종점간 지연에서 FEC coding 지연을 뺀 값이 된다. FEC coding 지연은 FEC encoding 지연과 FEC decoding 지연을 포함한다.

$$Da_i^t = \frac{\sum_{j=0}^{n_i-1} (T_{display}(j) - T_{sampling}(j))}{n_i} \quad (5)$$

$$Db_i^t = Da_i^t - FEC\ coding\ delay \quad (6)$$

주기 t 에서 reward와 penalty는 식 (1)과 식 (2)를 통하여 구할 수 있으며 그 값들은 네트워크 상황에 따라 매 RTCP 주기마다 달라진다. SCCRP는 현재의 네트워크 상태를 보다 효율적으로 반영하기 위해서 매 주기마다 R_i 와 P_i 를 식 (7)과 식 (8)을 이용하여 주기적으로 평활화(smoothing)시킨다.

$$R_i = \alpha \left(\frac{Lb_i^t}{La_i^t} \right) + (1 - \alpha) R_{i-1} \quad (7)$$

$$P_i = \alpha \left(\frac{Pa_i^t}{Pb_i^t} \right) + (1 - \alpha) P_{i-1} \quad (8)$$

R_i 을 이용하여 코덱 조합 i 를 사용할 때 얻을 수 있는 이득으로서 복구 후 패킷손실률의 예측치 Le_i 을 구하고, 손실로서 P_i 을 이용하여 코덱 조합 i 의 복구 후 종점간지연의 예측치 De_i 을 구한다.

$$Le_i = Lb_i^t / R_i \quad (9)$$

$$De_i = Db_i^t \times P_i \quad (10)$$

SCCRP 알고리즘의 목적은 패킷손실률을 HIGH 임계치와 LOW 임계치 사이로 유지시키는 것이며, 종점간지연이 최소인 코덱 조합을 선택하는 것이다. 즉, 다음의 두 조건을 만족하는 코덱 조합을 찾는 것이다. (그림 3)은 이 알고리즘의 pseudo-code를 보여준다.

- 조건 I : $A = \{j | LOW \leq Le_j \leq HIGH\}$, $0 \leq j \leq N_{cc} - 1$

패킷손실률이 LOW 임계치보다 작거나 HIGH 임계치보다 클 때 현재 FEC 부가정보량을 조절해야 하며 이를 위해서 현재의 코덱 조합을 변경해야 한다. 따라서 코덱 조합 j 의 복구 후 패킷손실률의 예측치 Le_j 가 LOW와 HIGH 사이에 있는 코덱 조합들의 집합 A 를 구한다. 이 때 N_{cc} 는 코덱 조합의 개수이다. 이 조건은 패킷손실률 면에서 코덱 조합을 변경함으로써 얻을 수 있는 이득(reward)을 고려하는 것이다.

- 조건 II : $\min_{k \in A} (De_k)$

집합 A 의 코덱-조합들 중 복구 후 종점간지연의 예측치 De_k 가 최소인 원소 k 가 다음의 코덱-조합으로 선택된다. 이 조건은 종점간지연 면에서 코덱 조합을 변경함으로써 영향을 받을 수 있는 손해(penalty)를 고려하는 것이다.

Procedure SCCRP_algorithm()

Begin Procedure

For (새로운 RTCP 패킷이 도착)

복구 전 패킷손실률과 복구 후 패킷손실률을 계산; /* 식 (3)과 식 (4) */

복구 후 종점간지연과 복구 전 종점간지연을 계산; /* 식 (5)와 식 (6) */

현재 코덱-조합의 reward와 penalty를 갱신; /* 식 (7)과 식 (8) */

If ((현재 패킷손실률 > HIGH) 또는 (현재 패킷손실률 < LOW))

/* 식 (9)와 식 (10) */

복구 후 패킷손실률의 예측치와 복구 후 종점간지연의 예측치를 구함;

/* 조건 I */

복구 후 패킷손실률의 예측치를 구하고 이 값이 LOW와 HIGH 임계치 사이에 있는 코덱-조합을 집합을 구함;

/* 조건 II */

위의 코덱-조합들중 복구 후 종점간지연의 예측치가 가장 작은 코덱 조합을 선택함;

End If

End For

End Procedure

(그림 3) SCCRP 알고리즘의 pseudo code

4. 실험 및 결과

SCCRP 알고리즘의 성능을 측정하기 위해서 실제로 인터넷 상에서 측정된 네트워크 트레이스를 이용하였다. 그러나 실제 수집된 인터넷 트레이스가 제한된 범위의 패킷손실률과 지연을 나타내므로 정확한 성능측정이 어려우므로 NS-2를 이용하여 다양한 네트워크 트래픽을 반영하는 트레이스를 수집하고 이를 실험하였다. 성능비교를 위하여 Bolot 알고리즘과 USF 알고리즘의 복구 후 패킷손실률과 복구 후 종점간지연을 비교하였다.

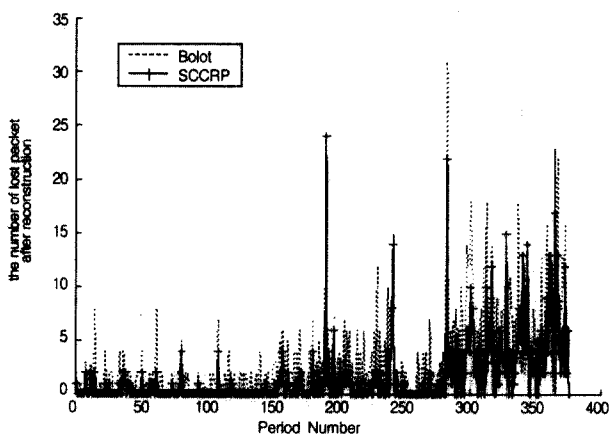
4.1 실측에 기반한 성능평가

인터넷 트레이스(Internet trace)를 위하여 실제로 부산대와 성심여대 사이의 인터넷망을 통하여 음성 패킷을 송수신하는 실험을 수행하였으며 RTCP 패킷을 이용하여 패킷 손실과 종점간지연을 측정하였다. 송신자와 수신자 사이의 평균 패킷손실률은 5.7%였고 평균지연시간은 101ms였다. 그리고 각 알고리즘의 성능을 실험할 때 HIGH와 LOW 임계치는 각각 5%와 3%로 설정하였고 RTCP 패킷은 5초 단위로 전송됨을 가정하였으며 평활계수(α)는 0.9로 설정하였다.

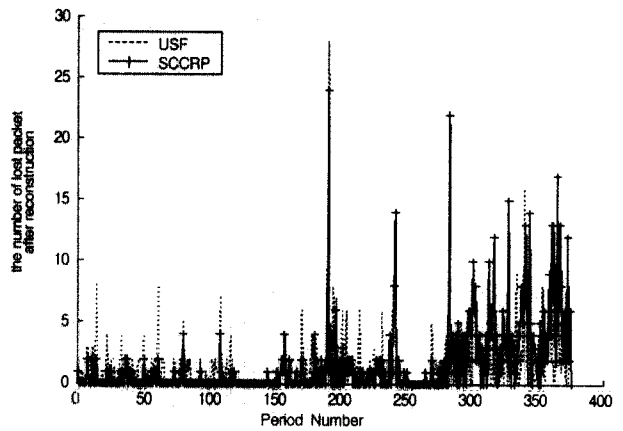
<표 4> 인터넷 trace 특성 및 실험 인자

인터넷 Trace 특성	송신자 / 수신자	부산대 / 성심여대
	패킷 크기	160 bytes (20ms)
	평균 패킷손실률	5.7%
	평균 종점간지연	101ms
실험 인자	HIGH 임계치	5%
	LOW 임계치	3%
	RTCP 주기	5초
	평활 계수 (α)	0.9

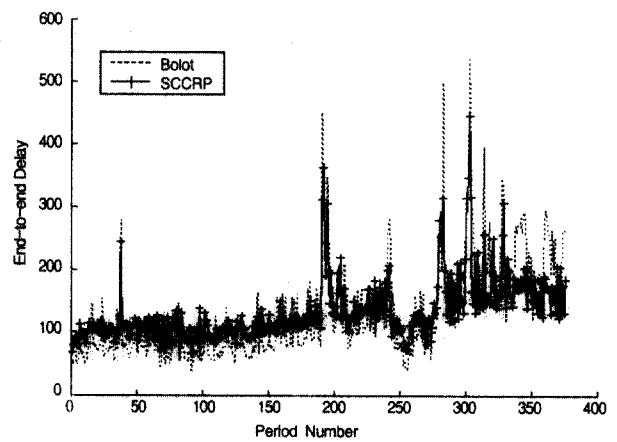
(그림 4)과 (그림 5)는 각 알고리즘을 적용했을 때 얻어지는 복구 후 패킷손실률을 보여준다. 그림에서 볼 수 있듯이 SCCRP의 복구 후 패킷손실률은 Bolot이나 USF에 비해 성능이 좋을 수 있다. (그림 6)와 (그림 7)은 복구 후 종점간지연을 비교한 것이다. SCCRP 알고리즘은 전체적으로 종점간지연을 다른 기법보다 더 낮게 유지한다. 특히, 그래프의 후반부에서(네트워크 상태가 좋지 못한 상태) 다른 알고리즘에 비해 종점간지연을 현저히 줄일 수 있다. <표 5>는 전체 실험 기간에서 패킷손실률이 HIGH 임계치를 초과한 주기의 비율을 보여준다. 표에서 볼 수 있듯이 SCCRP 알고리즘이 전체적으로 패킷손실률을 낮게 유지하였고 지연시간 임계치를 넘은 주기가 가장 작음을 알 수 있다.



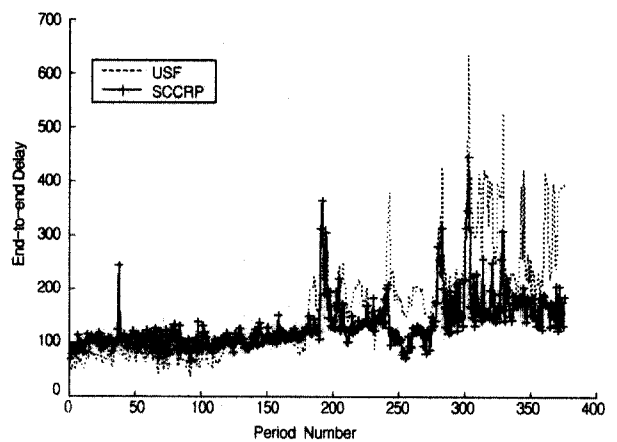
(그림 4) Bolot과 SCCRP의 복구 후 패킷손실량의 비교



(그림 5) USF과 SCCRP의 복구 후 패킷손실량의 비교



(그림 6) Bolot과 SCCRP의 복구 후 종점간지연 비교



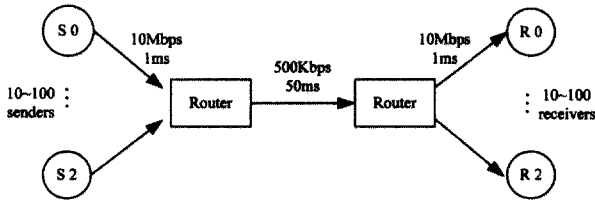
(그림 7) USF와 SCCRP의 복구 후 종점간지연 비교

<표 5> 실제 인터넷 trace를 이용한 각 기법의 성능 비교

	패킷손실률의 임계치를 넘은 주기의 비율	종점간지연의 임계치를 넘은 주기의 비율
Bolot 알고리즘	0.44	0.14
USF 알고리즘	0.25	0.32
SCCRP 알고리즘	0.19	0.09

4.2 NS-2를 이용한 성능평가

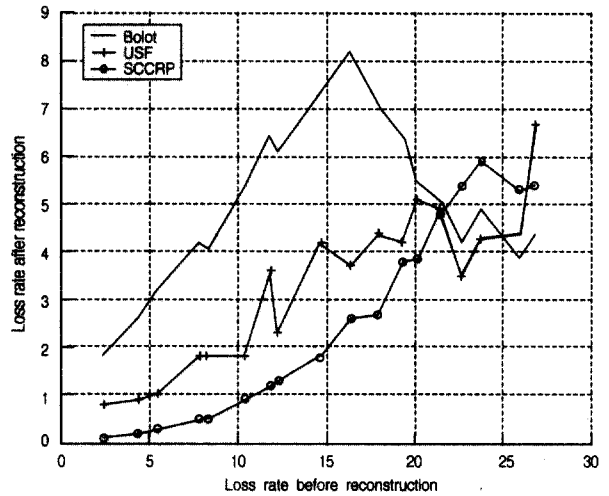
실제 실험에 의해서 측정된 인터넷 트레이스는 평균 패킷 손실률이 5.7%이고 평균 지연 시간이 101ms로 네트워크 상태가 비교적 양호하다고 할 수 있으며 실험 결과 각 기법의 성능이 크게 차이가 나지 않음을 알 수 있었다. 보다 정확한 실험을 위하여 Network Simulation Tool[19]인 NS-2를 이용하여 패킷손실률과 네트워크 지연이 다양하게 나타나는 트레이스를 생성하였다. (그림 8)는 실험에 사용된 NS-2의 simulation topology를 보여주는데 라우터사이의 bottleneck link의 대역폭은 500Kbps이고 전파지연은 50ms이며, 송신자 또는 수신자와 라우터사이는 10Mbps로 과잉밀집은 거의 없다고 가정하였다. 라우터에서는 다양한 큐길이를 가지는 RED queuing 기법을 적용하였다. 트래픽을 생성하기 위해서 여러 개의 TCP flow들과 하나의 UDP flow를 가정하고, UDP flow가 경험하는 패킷 손실과 지연을 트레이스로 기록하였다. 각 TCP flow의 패킷크기는 1000byte로 설정하였고, UDP flow의 패킷크기는 160byte로 설정하였고, 측정은 5분 동안 이루어졌다.



(그림 8) NS-2 토폴로지 구성

(그림 9)는 네트워크의 평균 패킷손실률에 따라 각 기법의 복구 후 패킷손실률의 변화를 보여준다. Bolot 알고리즘은 초기 설정된 reward 값이 고정되어 실제 네트워크 트래픽에 적용했을 때 네트워크 상태를 정확하게 반영하지 못한다. 또한,

코덱 테이블의 인덱스가 증감을 반복하는 순환현상이 발생하여 전체적으로 성능이 떨어진다. (그림 9)에서 볼 수 있듯이 복구 후 패킷손실률이 증가하다가 감소하는 패턴을 보이는데 이것은 순환현상 때문이다. <표 6>를 보면, 네트워크 패킷손실률이 16.3%일 때 코덱 테이블 인덱스가 증감을 반복함에 따라서 복구 후 패킷손실률이 큰 폭으로 등락을 반복한다. 따라서 평균 복구 후 패킷손실률이 높아진다. 그러나, 네트워크 패킷손실률이 25.9%일 때는 순환현상이 발생하지만 복구 후 패킷손실률의 등락의 폭이 작아 평균 복구 후 패킷손실률이 낮아진다.



(그림 9) 복구 후 패킷손실률의 비교

USF는 미리 추정된 reward 값을 사용하지 않고 RTCP의 실측정보를 기반으로 동작한다. 또한, 순환현상을 방지하기 위해 LOW 임계치를 벗어났을 때 바로 코덱 테이블의 인덱스를 감소시키지 않고 이전 복구 후 패킷손실률과 현재 복구 후 패킷손실률의 차이가 일정한 임계치 값을 넘을 경우만 코

<표 6> Bolot 알고리즘의 주기별 성능 측정 결과

주기 No.	네트워크 평균 패킷손실률 16.3%			네트워크 평균 패킷손실률 25.9%		
	복구 후 패킷손실률 (%)	평균지연 (ms)	코덱테이블 index	복구 후 패킷손실률 (%)	평균지연 (ms)	코덱테이블 index
10	12	399	2	2	462	4
11	2	366	1	1	503	3
12	11	374	2	6	505	4
13	2	372	1	1	505	3
14	14	366	2	6	543	4
15	0	450	3	0	455	5
16	13	381	2	6	495	4
17	3	370	1	3	505	3
18	12	372	2	11	529	4
19	0	373	1	2	481	5
20	15	378	2	3	488	4

<표 7> USF 알고리즘의 주기별 성능 측정 결과

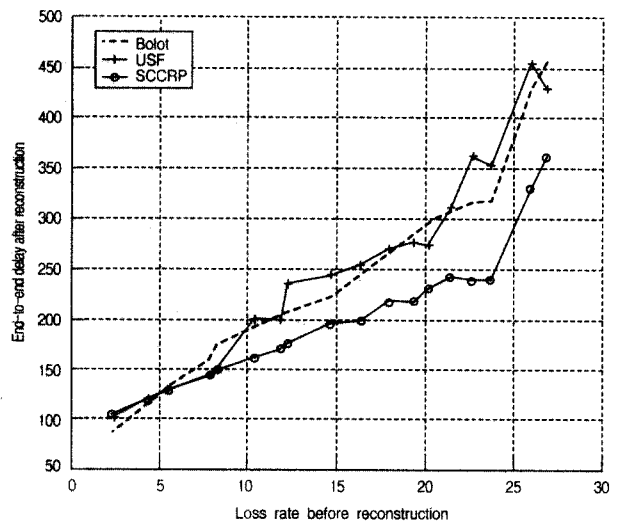
주기 No.	네트워크 평균 패킷손실률 16.3%			네트워크 평균 패킷손실률 25.9%		
	복구 후 패킷손실률 (%)	평균지연 (ms)	코덱테이블 index	복구 후 패킷손실률 (%)	평균지연 (ms)	코덱테이블 index
10	0	434	3	0	589	5
11	2	420	3	9	480	4
12	0	419	3	1	609	5
13	11	330	2	4	446	4
14	5	444	3	3	628	5
15	4	454	4	0	627	5
16	0	575	5	2	456	4
17	1	456	4	2	479	4
18	9	436	4	7	439	4
19	0	565	5	1	608	5
20	2	457	4	0	625	5

<표 8> SCCRP 알고리즘의 주기별 성능 측정 결과

주기 No.	네트워크 평균 패킷손실률 16.3%			네트워크 평균 패킷손실률 25.9%		
	복구 후 패킷손실률 (%)	평균지연 (ms)	코덱테이블 index	복구 후 패킷손실률 (%)	평균지연 (ms)	코덱테이블 index
10	2	352	1	4	352	1
11	2	318	1	10	329	4
12	0	335	4	2	370	4
13	2	323	1	11	329	1
14	6	314	1	3	418	6
15	2	332	2	2	417	6
16	2	337	1	14	339	1
17	3	319	1	3	409	6
18	1	333	4	0	428	6
19	0	331	1	3	376	1
20	2	332	1	1	377	6

덱 테이블 인덱스를 변경한다. 따라서, Bolot 알고리즘보다 더 높은 성능을 보인다. <표 7>은 코덱 테이블의 인덱스와 복구 후 패킷손실률의 순환현상이 줄었음을 보여준다. 또한, 네트워크 패킷손실률이 25.9%일 때가 16.3%일 때 보다 더 높은 테이블 인덱스 값을 가짐으로써 전체적으로 낮은 복구 후 패킷손실률을 유지함을 보인다.

SCCRP는 코덱 테이블의 인덱스를 1씩 증가하는 것이 아니라 증감폭을 다양하게 결정할 수 있기 때문에 가변적인 네트워크 상태에 더 적응적으로 대처할 수 있다. <표 8>에서 볼 수 있듯이, 이 기법은 네트워크 패킷손실률이 16.3%일 때 필요한 만큼 테이블 인덱스를 증감시킴으로써 복구 후 패킷손실률을 낮게 유지한다. 그러나, 네트워크 패킷손실률이 높을 때 (25.9%), 폭이 넓은 순환현상이 발생하여 복구 후 패킷손실률이 다소 증가한다.



(그림 10) 복구 후 종점간지연의 비교

(그림 10)은 네트워크 평균 패킷손실률에 따라 복구 후 종점간지연을 보여준다. SCCRP는 종점간 지연을 고려하기 때문에 다른 알고리즘보다 현저히 종점간지연을 줄일 수 있다. 패킷손실률이 적을 때 상대적으로 다른 알고리즘보다 높은 이유는 종점간지연이 허용하는 범위에서 부가정보를 많이 보내기 때문이다. Bolot과 USF 알고리즘들은 종점간지연을 고려하지 않고 코덱 조합을 선택함으로써 코딩 지연이 높은 코덱을 선택할 가능성이 높기 때문에 종점간지연은 증가한다.

5. 결론 및 향후 연구

기존의 FEC 기반 에러 제어 알고리즘은 부가 정보를 조절할 때 패킷손실률만을 고려하였기 때문에 인터넷 전화와 같은 종점간지연에 민감한 대화식 응용에 적용하기는 적합하지 못했다. 따라서 본 논문은 인터넷 전화의 효율적인 통화 품질을 위하여 패킷손실률은 물론 종점간지연을 고려한 에러 제어 기법 SCCRP를 제안하였다. 실험 결과, Bolot 알고리즘은 고정된 reward값과 순환현상으로 인하여 전체적으로 가장 낮은 성능을 보이고, USF는 순환현상보다는 네트워크 상태가 불안정 또는 안정한지에 따라 성능차를 보였다. 제안된 SCCRP는 종점간지연을 고려하고 코덱 테이블 인덱스의 증감폭을 다양하게 결정함으로써 가장 높은 성능을 유지하였다. 향후에는 코덱 조합의 전송률까지 고려한 에러 제어 기법을 연구하고자 한다. 패킷손실률이나 종점간지연을 줄이기 위해서 부가정보를 증가시키는 것은 네트워크의 과부하를 초래할 수 있으므로 penalty에 전송률이 고려되어야 한다.

참고 문헌

- [1] S. Blake, D. Blake, M. Carlson, et al, "An architecture for differentiated service," RFC 2475, IETF, Dec., 1998.
- [2] J. Wroclawski, "Specification of the controlled-load network element service," RFC 2211, IETF, Sept., 1997.
- [3] S. Shenker, C. Partridge, and R. Guerin, "Specification of guaranteed quality of service," RFC 2212, IETF, Sept. 1997.
- [4] Ramachandran Ramjee, Jim Kurose, Don Towsley, and Henning Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide-area networks," in Proc. of IEEE Infocom, Toronto, Canada, June, 1994.
- [5] Sue B. Moon, Jim Kurose, and Don Towsley, "Packet audio playout delay adjustment : performance bounds and algorithms," ACM/Springer Multimedia Systems, Vol.5, No.1, Jan., 1998.
- [6] D. Sisalem and H. Schulzrinne, "The loss-delay based adjustment algorithm : A TCP-friendly adaptation scheme," Proc. of NOSSDAV, Cambridge, England, July, 1998.
- [7] R. Rejaie, M. Handley, and D. Estrin, "RAP : An end-to-end rate-based congestion control mechanism for realtime streams in the Internet," Proc. of IEEE Infocom, Mar. 1999.
- [8] J. L. Ramsey, "Realization of optimum interleavers," IEEE Transactions on Information theory, IT-16, May, 1970.
- [9] D. J. Goodman, G. B. Lockhart, O. J. Wasem, and W. C. Wong, "Waveform substitution techniques for recovering missing speech segments in packet voice communications," IEEE Transactions on Acoustics, Speeches, and Signal Processing, ASSP-34(6), Dec., 1986.
- [10] S. Floyd, V. Jacobson, S. McCanne, et al. "A reliable multicast framework for light-weight sessions and applications level framing," IEEE/ACM Transactions on Networking, Dec., 1997.
- [11] Nachum Shacham and Paul McKenney, "Packet recovery in high-speed networks using coding and buffer management," in Proc. of IEEE Infocom, San Francisco, California, June, 1990.
- [12] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," ACM Computer Communication Review, Vol.27, April, 1997.
- [13] J. C. Bolot, and A. V. Garcia, "Control mechanisms for packet audio in the internet," in Proc. of IEEE Infom, San Fransisco, California, Mar., 1996.
- [14] C. Padhye, K. J. Christensen and W. Moreno, "A new adaptive FEC loss control algorithm for voice over IP applications," in Proc. of IEEE IPCC00, 2000.
- [15] C. Perkins, O. Hodson, and Hardman, "A Survey of Packet Loss Recovery Techniques for Streaming Audio," IEEE Network, Vol.12, No.5, pp.40-48, September, 1998.
- [16] J. Rosenberg and H. Schulzrinne, "An RTP payload format for generic forward error correction," IETF Audio/Video Transport Working Group, RTC2198, July, 1998.
- [17] V. Hardman, M. A. Sasse, M. Handley and A. Watson, "Reliable audio for use over the Internet," in Proceedings of INET'95, 1995.
- [18] J.-C. Bolot, A. Vega-Garcia, "The case for FEC based error control for packet audio in the Internet," to appear in ACM/Springer Multimedia Systems, 1999.
- [19] ns - Network Simulator. See <http://www-mash.cs.berkeley.edu/ns/ns.html>
- [20] Myoung-Kyoung Ji, Tae-Uk Choi, et al, "Selecting an audio redundancy codec combination for error control in Internet telephony," in Proc. of ICPADS2001, KyongJu, Korea, June, 2001.
- [21] Myoung-Kyoung Ji, Tae-Uk Choi, et al, "Adaptive FEC error control algorithm using loss and delay for Internet telephony," in Proc. of ICME2001, Tokyo, Japan, August, 2001.

최 태 욱

e-mail : tuchoi@melon.cs.pusan.ac.kr

1997년 동의대학교 전산통계학과 졸업(학사)

1999년 부산대학교 대학원 전자계산학과
졸업(이학석사)

2002년 부산대학교 대학원 전자계산학과
박사 수료

관심분야 : VoIP, 멀티미디어, QoS

정 기 동

e-mail : kdchung@melon.cs.pusan.ac.kr

1973년 서울대학교 졸업(학사)

1975년 서울대학교 대학원 졸업(이학석사)

1986년 서울대학교 대학원 졸업(이학박사)

1990년~1991년 MIT대학 교환 교수

1995년~1997년 부산대학교 전자계산소 소장

1999년~2001년 부산대학교 BK21 사업 단장

1978년~현재 부산대학교 전자계산학과 교수

관심분야 : 병렬처리, 멀티미디어