

인터넷 폰에서 Synchronized overlap-add 알고리즘을 이용한 전송지연 보상 기법

(Concealment of Propagation Delay using Synchronized overlap-add Algorithm in Internet Phone)

남재현[†] 이정태^{**}
(Jae-Hyun Nam) (Jung-Tae Lee)

요약 인터넷전화 서비스는 저렴한 가격과, 타 서비스와 통합 및 가치부가(Value Added)면에서 기존의 전화에 비해 많은 장점을 가지고 있으나, 상대적으로 낮은 음질로 인하여 사용자의 요구를 만족시키지 못하고 있다. 이것은 현재 인터넷은 best-effort형 패킷 전달 서비스만을 제공하고 있기 때문에 전송지연, 패킷손실, 지터 등을 보장할 수 있는 방법이 없기 때문이다. 본 논문에서는 인터넷전화에서 패킷손실이나 전송지연으로 인한 음질 저하문제를 SOLA 알고리즘을 이용해 보완하였다. SOLA 알고리즘은 시간축변환(Time Scaled Modification) 기법중의 하나로써 음성신호가 가지는 중요한 스펙트럼 정보는 그대로 유지하면서 단지 발음 속도만을 변환시키는 기법이다. 본 논문에서는 송신측에서 패킷을 전송하면 수신측에서는 수신 패킷에 SOLA 알고리즘을 적용하여 수신 패킷을 사람이 인지하지 못하는 수준에서 확장하여 전송지연으로 인한 패킷손실을 감소시킨다. 시뮬레이션 결과 전송지연으로 인한 패킷 손실 확률이 상당히 감소되었고 음질 또한 상당히 개선되었다.

Abstract Internet telephony service is very cheap and very easy to introduce the value-added service than the POTS, but is difficult to the QoS of telephone service. The existing Internet typically offers "best effort" services only, which do not make any commitment about delay, packet loss and jitter. This paper compensates the low quality of the speech for packet loss or delay using SOLA algorithm in Internet phone. SOLA algorithm is a popular technique for Time Scale Modification of speech and audio signal. In the proposed algorithm, the receiver expands the received packet under reasonable threshold, and hence compensates the QoS of speech. From the simulation, this algorithm can conceal packet loss considerably, and is also improved the quality of the speech.

1. 서론

인터넷 인구의 급속한 증가로 인해 인터넷을 통해 다양한 트래픽을 전송하고자 하는 요구가 증가하고 있다. 지금까지 인터넷은 전자우편이나 Web과 같은 데이터 트래픽을 중심으로 발전해 왔지만, 사용자가 점차 원격 교육, 원격회의 등과 같은 실시간 트래픽의 전달을 필요로 하는 응용(Applications)을 요구함에 따라, 인터넷도 이제 음성이나 화상과 같은 실시간 트래픽을 수용하는

방향으로 나아가고 있다.

이러한 변화는 네트워크구조에도 많은 영향을 끼쳐, 공중망(Public networks)의 중심이 회선교환망(Circuit switched networks)에서 패킷교환망(Packet switched networks)으로 바뀌고 있고, 결국 PSTN이 IP 네트워크에 통합되는 결과를 가져 올 전망이다. 이러한 통합은 인터넷과 같은 IP 네트워크를 통해 음성, 팩스 등을 전송할 수 있음을 의미한다. 이와 같이 인터넷에는 새로운 서비스가 끊임없이 추가되고 있는데, 그 중 대표적인 것이 인터넷을 통하여 전화서비스를 제공하는 인터넷전화(Internet Phone) 서비스이다.

인터넷전화 서비스는 저렴한 가격과, 타 서비스와 통합 및 가치부가(Value Added)면에서 기존의 전화에 비해 많은 장점을 가지고 있으나, 상대적으로 낮은 음질로

[†] 비회원 : 동주대학 컴퓨터정보통신계열 교수
jhnam@dongju.ac.kr

^{**} 종신회원 : 부산대학교 컴퓨터공학과 교수
jilee@hyowon.cc.pusan.ac.kr

논문접수 : 2000년 10월 4일

심사완료 : 2001년 8월 2일

인하여 사용자의 요구를 만족시키지 못하고 있다[1]. 이것은 현재 인터넷은 전송되는 트래픽의 형태에는 상관 없이 모든 트래픽이 차별없이 가능한 빠르고 많은 대역폭을 할당하는 방식으로 패킷이 전달되는 Best-effort형 패킷전달 서비스만을 제공하고 있기 때문이다. 이러한 패킷전달방식은 실시간 응용 서비스가 요구하는 엄격한 QoS 보장을 만족시킬 수 없다. 즉, 근본적으로 Best-effort서비스만을 지원하는 인터넷을 통하여 음성을 전달하는 데에는 지연(Delay), 패킷손실(Packet loss) 및 지터(Jitter) 등의 통화 품질 열화요인들에 대한 문제점이 해결되어야만 toll급의 음성품질의 얻을 수 있다. 각각의 인자에 대한 음성품질의 영향을 살펴보면 다음과 같다.

첫째, 지연은 패킷을 전달할 때 종단간에 걸리는 시간으로, 인터넷전화에서 상대방과 통화를 할 경우, 지연이 너무 크면 원활한 대화가 어려워진다. ITU-T에서는 종단간의 전화통화에서 단방향 지연시간을 400ms 이내로 권고하고 있다[2]. 그러나 그림 1에 제시된 바와 같이 인터넷전화에서 toll급의 통화품질의 얻기 위해서는 단방향 지연시간이 100ms 이내가 되어야 한다[3].

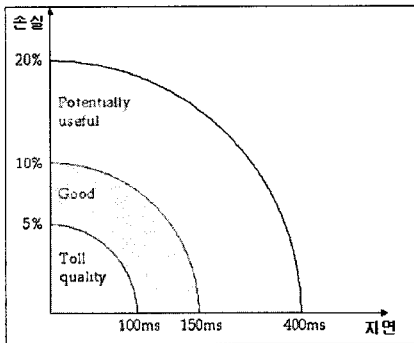


그림 1 지연시간과 패킷손실률에 따른 음성품질

둘째, 패킷손실은 라우터 등에서의 폭주(Congestion)가 발생되거나 음성과 같은 실시간 응용에서 패킷이 수신측에서 플레이되는 시점보다 늦게 도착할 경우 발생되는데, 패킷손실이 증가되면 통화품질이 저하된다. 일반적으로 인터넷상에서의 패킷 손실은 0~20%까지 나타나고 있지만 원만한 통화 품질을 얻기 위해서는 10% 이내가 되어야 한다[4].

마지막으로 지터는 전송지연시간의 편차를 의미하는데, 이 값이 너무 크면 수신측에서 음성이 일정한 간격으로 재생되지 못하고, 일그러지는 현상이 발생하므로 패킷손실과 마찬가지로 음성품질을 저하시킨다[5]. 표 1

은 지터에 따른 음성품질의 정도를 보여주고 있는데, 지터가 75ms 이내이어야 만족할 만한 수준의 품질을 얻을 수 있음을 알 수 있다[6].

표 1 지터에 따른 음성품질

Quality	Jitter
Perfect	0ms
Good	75ms
Medium	125ms
Poor	225ms

본 논문에서는 현재 사용되고 있는 인터넷의 특성을 살펴보기 위하여 [7]에서 제시된 지연시간과 지터, 그리고 패킷손실률을 참조하였다. [7]에서 제시된 측정 결과를 살펴보면 그림 2와 같이 국내 망의 경우 인터넷전화는 toll급의 통화품질의 얻을 수 있지만 국제 망의 경우에는 전송지연 뿐만 아니라 패킷손실률이 상당히 크게 나타나고 있기 때문에 아직 품질이 만족스럽지 못함을 알 수 있다.

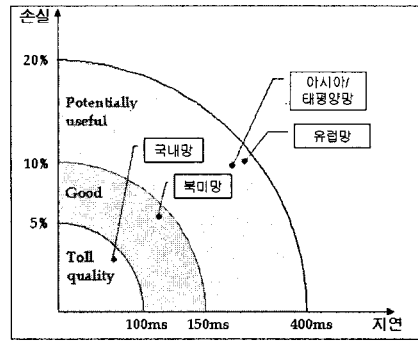


그림 2 지연시간과 패킷손실에 따른 음성품질 실측 결과

인터넷 전화의 품질을 저하시키는 요인 중의 하나인 패킷 손실 문제를 해결하기 위해 많은 연구들이 진행되어 왔는데 크게 패킷 손실을 보상하기 위해 송수신 단말기 측에서 손실된 패킷을 보상하는 기법과 전송지연으로 인한 패킷 손실을 줄이기 위한 버퍼링 기법 등이 있다[8, 9]. 이러한 기법은 연속적인 패킷 손실이 발생할 경우 손실패킷을 보상하더라도 음질이 상당히 저하되고, 전송지연으로 인한 패킷 손실을 줄이기 위해 버퍼링 기법을 사용할 경우 종단간의 지연이 증가함에 따라 음성품질이 저하된다.

본 논문에서는 인터넷전화에서 전송지연으로 인한 패

킷 손실 때문에 발생하는 음질 저하문제를 SOLA (Synchronized Overlap and Add) 알고리즘[10]을 이용해 보완하고자 한다. 즉, 수신된 각 프레임의 크기를 SOLA 알고리즘을 이용하여 일정한 크기로 확장시켜서 전송지연으로 인한 패킷 손실을 줄이고자 한다.

이 글의 구성은 2장에서는 관련연구에 대해 살펴보고, 3장에서는 SOLA를 이용한 패킷 손실 보상 기법을 제시하였고 4장에서는 시뮬레이션을 통해 다른 기법과의 MOS(Mean Opinion Score)를 비교하여 제시된 기법의 성능을 분석하였으며, 끝으로 5장에서는 결론을 제시하였다.

2. 관련 연구

오디오 패킷이 손실될 경우 이를 복원하는 방법에는 그림 3과 같이 송신측 보상기법과 수신측 보상기법이 있다[8].

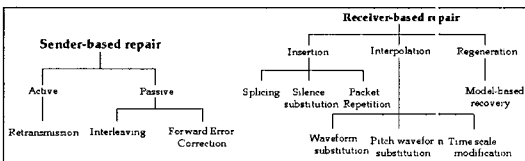


그림 3 패킷 손실 보상 기법

2.1 송신측 보상기법

송신측 보상기법(Sender-based repair technique)은 Active 기법과 Passive 기법으로 분류할 수 있는데, Passive 기법은 다시 순방향 에러 수정(Forward Error Correction)과 인터리빙(Interleaving) 기법으로 나눌 수 있다.

순방향 에러 수정 기법은 하나의 스트림에 손실된 패킷을 복구할 수 있는 정보를 추가시켜서 전송하는 기법이다. 이러한 기법은 전송되는 미디어 종류와는 무관하게 적용시킬 수 있고 상대적으로 적은 계산시간으로 손실 복구 패킷을 생성할 수 있지만 오버헤드가 너무 커서 대역폭이 증가되는 단점이 있다.

인터리빙 기법은 오디오 데이터내의 패킷들을 일정한 간격으로 재배치하여 전송한 후 수신측에서 다시 원래의 순서대로 복원시키는 기법이다. 이 기법은 패킷 손실의 영향을 분산시켜준다. 예를 들면 한 패킷은 4 유니트(unit)로 구성되고 유니트 길이가 5ms, 패킷의 길이가 20ms일 경우 첫 번째 패킷은 1, 5, 9, 13 유니트로 구성되고 두 번째 패킷은 2, 6, 10, 14 유니트로 구성된다. 이 기

법은 대역폭의 증가가 필요하지 않은 장점이 있지만 상대적으로 높은 지연시간(latency)을 지니는 단점이 있다.

2.2 수신측 보상기법

수신측 보상기법(Receive-based repair technique)은 손실된 패킷에 대해 원래의 패킷과 유사한 패킷을 만들어서 대체시키는 기법이다. 이것은 오디오 신호의 경우 서로 이웃한 패킷들은 유사한 특성을 지니고 있다는 점을 이용한 것이다. 이 기법은 크게 세 가지로 나눌 수 있다.

삽입(Insertion) 방법은 손실된 패킷을 묵음이나 잡음, 또는 손실 직전의 패킷으로 대체하는 기법이다. 이 기법은 구현은 간단하나 음질이 상당히 떨어진다.

보간(Interpolation) 기법은 손실된 패킷과 유사하다고 예상되는 대치 패킷을 생성하기 위해 손실 주위에 있는 패킷과 패턴비교나 보간을 수행한다. 이러한 기법은 삽입방법보다 계산량은 많지만 신호의 변화 특성을 고려할 수 있기 때문에 삽입방법보다 나은 음질을 가진다. 본 논문에서 사용한 SOLA 알고리즘은 이 부류에 속한다.

재생(Regeneration) 기법은 손실된 패킷의 앞과 뒤 패킷의 정보를 이용하여 손실 패킷을 대체할 패킷을 생성한다. 이 기법은 삽입이나 보간 기법보다 계산량은 더 많이 요구되지만 상대적으로 더 나은 음질을 얻을 수 있다.

이 밖에도 Adaptive playout buffering algorithm[9]이 있는데 이 기법은 각 talkspurt에 대한 playout delay D 를 선택하여 각 talkspurt의 playout 시점을 $t_i + D$ 로 조정하는 기법이다. 여기서 t_i 는 패킷의 생성시간, D 는 하나의 talkspurt내의 모든 패킷들의 생성시간과 playout 시간간의 차이를 말한다. D 가 증가하면 패킷들이 늦게 도착하여 손실로 처리되는 부분은 줄어들지만 중단간에 발생하는 지연시간은 커지게 된다.

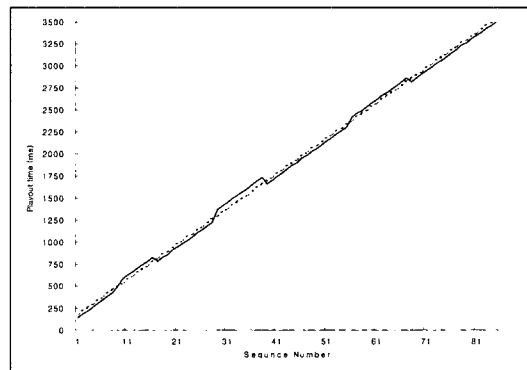


그림 4 패킷들의 정상적인 playout 시간과 도착시간

그림 4에서는 송신측에서 주기적으로 패킷을 생성하여 네트워크를 통해 전송하면 전송되는 도중 각 패킷들은 임의의 시간만큼 지연되어서 수신측에 도달되어 playout되는 형태를 보여주고 있다. 그림 4에서 점선은 송신측에서 일정한 간격으로 패킷을 전송할 경우 수신측에서 정상적으로 playout되는 시점이고 실선은 해당 패킷이 네트워크를 통해 전송될 때 지연(delay)를 겪어 실제로 수신측에 도달되었을 때를 의미한다. 그림 4에서 보는 바와 같이 패킷 번호 11 ~ 20, 32 ~ 41, 61 ~ 72는 네트워크를 통해 정상적으로 수신측에 도달하였지만 자신의 playout 시점이 지난 후에 수신측에 도달한 경우이다. 이러한 경우 해당 패킷은 네트워크를 통해 전송할 경우에는 손실이 발생되지 않았지만 실제로 수신측 application에서 자신의 playout 시점이 지났기 때문에 손실로 간주된다. 즉, 수신측 application은 수신된 패킷 중 첫 번째 패킷이 playout되면 이후 패킷부터는 일정한 시간 간격으로 계속적으로 playout이 수행되어야 하기 때문에 해당 시점 이후에 도달한 패킷은 playout되지 못하고 손실로 처리된다.

예를 들면 그림 5와 같이 각 talkspurt가 n개의 패킷들로 구성되어 있다면 talkspurt내의 각 패킷들은 네트워크를 통해 전송될 경우 서로 다른 전송 지연을 겪게 되어 수신측에 $a_k^1, a_k^2, \dots, a_k^n$ 시간에 도달한다. 수신측 application에서 적당한 버퍼링을 수행하여 talkspurt내의 첫 번째 패킷의 playout time을 p_k^1 로 설정하면 이후 패킷들은 송신측에서 전달한 시간간격으로 playout time이 설정된다. 이 때 수신측에 a_k^4 시간대에 도달한 패킷은 수신측 application에서 playout 되는 시점보다 늦게 도달했기 때문에 수신측 application에서는 a_k^4 시간대에 도달한 패킷을 playout하지 못하고 손실로 처리한다.

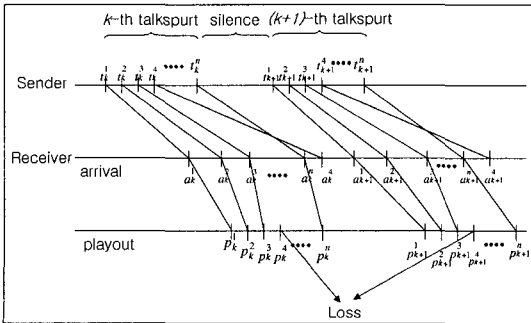


그림 5 talkspurt내의 각 패킷들의 도착과 playout 시간

따라서 Adaptive playout buffering algorithm은 지연 지터(delay jitter)로 인한 패킷 손실을 줄이기 위해 수신측 호스트는 수신된 패킷의 playout 시간을 일정한 시간동안 지연시킨다. 이러한 알고리즘은 손실과 지연 시간사이에 최적으로 동작할 수 있는 버퍼링 시간을 계산하여 최적 지점에 가깝게 playout delay를 조정한다. 각 패킷의 playout 시간은 다음과 같이 계산된다[9].

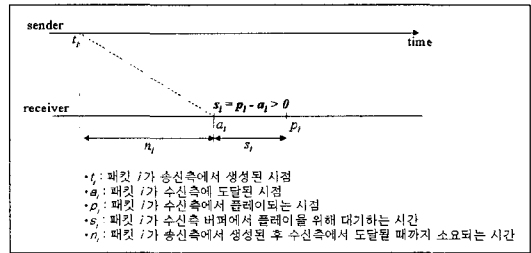


그림 6 패킷 i의 생성, 수신 및 playout 시간

패킷 i의 playout시점은 패킷이 talkspurt내의 첫번째 패킷인지 아닌지를 고려하여 두가지 경우로 나누어진다. 즉, 패킷 i가 talkspurt내의 첫번째 패킷일 경우 playout 시점은 $p_i = t_i + \hat{d}_i + 4 \hat{v}_i$ 로 계산된다. 여기서 \hat{d}_i 는 중단간의 평균 지연시간을 나타내고 \hat{v}_i 는 중단간의 지연 편차(variation)를 의미한다.

패킷 i가 talkspurt내의 첫번째 패킷이 아닐 경우 playout 시점은 $p_i = p_i + t_i - t_i$ 로 계산된다. 그리고 \hat{d}_i 와 \hat{v}_i 는 다음과 같이 계산된다[9].

$$\hat{d}_i = \alpha \times \hat{d}_{i-1} + (1 - \alpha) \times n_i$$

$$\hat{v}_i = \alpha \times \hat{v}_{i-1} + (1 - \alpha) \times \left| \hat{d}_{i-1} - n_i \right|$$

where $\alpha = 0.998002$

3. SOLA를 이용한 패킷 손실 보상 기법

음성은 인간의 의사 전달에 있어서 가장 중요한 수단의 하나이므로 정보통신 기술의 발달과 함께 음성 신호 처리에 대한 연구도 매우 활발히 진행되고 있다. 대표적인 연구분야중의 하나인 시간축 변환(Time Scaled Modification) 기법은 음성신호가 가지는 중요한 스펙트럼 정보는 그대로 유지하면서 단지 발음 속도만을 변환시키는 기법이다. 이러한 시간축 변환 방법에는 단구간 Fourier해석에 의한 방법[10]과 정현파 모델에 의한 방법[11], SOLA 방법[12], Pitch Synchronized Overlap and Add(PSOLA) 방법[13] 등이 있다.

이들 시간축 변화 방법 중 SOLA 방법은 실시간 처리가 가능하면서 비교적 고음질의 변환 신호를 얻을 수 있기 때문에 본 논문에서는 SOLA를 이용하여 전송 지연으로 인한 패킷 손실을 줄일 수 있는 방법을 제시하였다.

SOLA 기법은 음성신호의 시간축 변환을 위해 프레임 단위의 신호들을 정규화된 상호 상관 함수(normalized cross-correlation function)를 이용하여 축소 또는 확장하고 동기를 맞춘 후 중첩 가산(cverlap and add)하는 방법을 사용한다.

입력 프레임 x 에 변환 비율 α 를 적용하여 시간축으로 확장 또는 축소 변환된 프레임 y 를 얻을 경우 수행되는 과정은 그림 7과 같다.

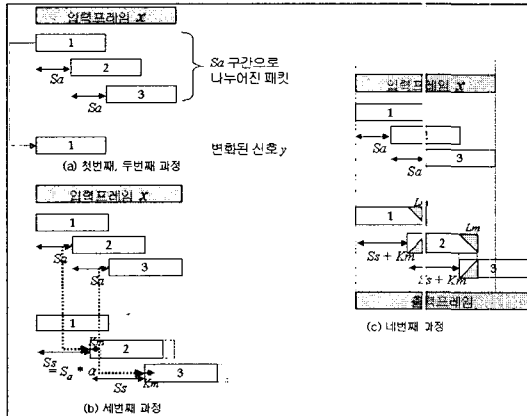


그림 7 SOLA 수행 과정

첫번째 과정은 입력 프레임 x 를 S_a 간격으로 시간축 변환에 이용할 패킷들로 나눈다. 두번째 과정은 S_a 간격으로 나눈 패킷 중 첫번째 패킷(1번 패킷)을 변환 프레임 y 의 첫번째 패킷으로 가져온다. 세번째 과정은 다음 패킷(2번 패킷)을 가져와서 y 에 S_s 간격으로 패킷을 위치시킨 후 서로간의 동기가 최대한 일치하는 지점을 얻기 위해 상호 상관 함수를 이용하여 동기가 일치하는 지점 K_m 을 찾는다. 마지막으로 동기가 일치하는 지점 $(S_s + K_m)$ 으로 재배치한 후 중첩되는 구간 L_m 의 값을 중첩 가산한다. 합성과정은 매 패킷당 이루어지고 새로운 패킷이 이전 과정에서 합성된 프레임 y 에 더해진다. 이 때 S_a 와 S_s 사이에는 $S_s = \alpha S_a$ 라는 관계가 성립한다. 따라서 그림 8에서와 같이 α 가 1보다 클 경우 확장의 의미를 가지고 1보다 적을 경우에는 축소한다는 의미를 지닌다.

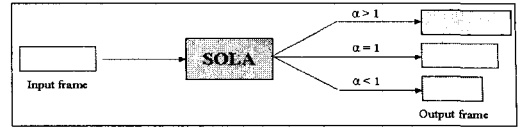


그림 8 변환비율에 대한 SOLA 적용 결과

동기가 일치하는 위치 K_m 은 $y(mS_s + j)$ 와 $x(mS_a + j)$ 사이의 정규화된 상호 상관 함수를 통해 얻어진다.

$$R_m(k) = \frac{\sum_{j=0}^{L-1} y(mS_s + k + j)x(mS_a + j)}{\left[\sum_{j=0}^{L-1} y^2(mS_s + k + j) \sum_{j=0}^{L-1} x^2(mS_a + j) \right]^{1/2}}$$

여기서 $R_m(k)$ 는 m 번째 패킷에서 정규화된 상호상관 함수를 나타내고, L 은 $y(mS_s + j)$ 와 $x(mS_a + j)$ 사이에서 중첩되는 샘플 수이다. $R_m(k)$ 를 최대화시키는 위치를 K_m 이라고 할 때, $x(mS_a + j)$ 는 $y(mS_s + K_m + j)$ 에 다음과 같이 재배열되어 중첩 가산된다.

$$y(mS_s + K_m + j) = (1 - f(j))y(mS_s + K_m + j) + f(j)x(mS_a + j), \quad 0 \leq j \leq L_m - 1$$

$$y(mS_s + K_m + j) = x(mS_a + j), \quad L_m \leq j \leq N - 1$$

여기서 L_m 은 두 신호의 중첩가산구간을 나타내고 $f(j)$ 는 가중치 함수로 $0 \leq f(j) \leq 1$ 의 값을 갖는다.

본 논문에서는 전송지연으로 인한 패킷 손실을 줄이기 위해 수신측에 전송된 프레임에 변환비율 $\alpha = 1.25$ 로 SOLA 알고리즘을 적용하여 프레임의 크기를 확장하였다. [14]에 제시된 결과를 보면 변화비율 α 가 0.75~1.5 사이일 때 90%이상의 인식률을 나타내고 있다. 하지만 시뮬레이션시 수신 프레임에 α 값을 다양하게 적용시켜 본 결과 α 가 1.25가 될 때 사람이 인지하지 못하는 수준으로 나타났다. 본 논문에서 제시한 SOLA 기법을 이용한 playout 시간 보상 알고리즘은 그림 9와 같다.

송신측에서 샘플링된 프레임에 전송하면 수신측에서는 수신프레임에 SOLA 알고리즘을 적용하여 수신프레임의 크기를 사람이 인지할 수 없을 수준까지 확장한다. 확장 후 발생하는 playout 지연시간을 계속 축적하여 전송지연으로 인한 패킷 손실을 감소시키는데 사용한다. 원음에 비해 과다하게 확장되는 것을 방지하기 위해 playout 지연시간의 상태(Playout_time - $(t_i + n_i) > 2 * \max(n_i)$)에 따라 패킷에 대한 SOLA 알고리즘 적용 여부를 결정한다. 재전송 타임아웃(retransmit timeout)은 저속 출발 라운드(slow start round)에서 계산된 재전송 타임아웃이 다음 라운드의 실제 RTT보다 적거나 같을 경우 불필요한 재전송이 발생되기 때문에, 이를 방지하기 위해 각 라운드에 대한 재전송 시간을 두 배로 설정한다[15]. 따라서 본 논문에서는 전송지연으로 인한

```

First_packet_playout_time = First packet arrive time +
                           SOLA_computation time + buffering time
Playout_time = First_packet_playout_time + First_packet_play_time
                                                    /*초기치*/
For (i)
{ i-th Packet Sender time =  $t_i$  ; i-th Packet Propagation time =  $n_i$ 
  If ( $t_i + n_i < \text{Playout\_time}$ ) Then
    If ( $\text{Playout\_time} - (t_i + n_i) > 2 \times \max(n_i)$ ) Then
      Normal packet playout
      Playout_time = Playout_time + normal packet play time
                                                    /*증가치*/
    Else
      SOLA_Expanded packet playout
      Playout_time = Playout_time + SOLA_Expanded packet play time
                                                    /*증가치*/
  End
Else
  Packet loss
End
}
    
```

그림 9 SOLA 기법을 이용한 playout 시간 보상 알고리즘

패킷 손실을 최대한 줄이기 위해 각 프레임의 전송시간 중 최대 시간의 두 배($2 * \max(n_i)$)로 설정하여 충분한 playout 지연시간을 가질 수 있도록 하였다.

그림 10에서는 각 프레임이 수신측에 수신된 후 playout되는 과정을 나타내었다. 도착 즉시 playout할 경우(case 1) 5번, 9번, 10번 프레임이 손실되지만 2 프레임만큼 지연시킨 후 playout할 경우(case 2) 손실되는 프레임은 없다. 하지만 이 경우 2 프레임 크기만큼 중단간에 지연이 발생된다.

수신측에 도착하는 모든 프레임에 대해 SOLA 알고리즘을 적용할 경우(case 3) 원음에 비해 과다하게 확장되어 전체 중단간의 지연시간이 증가되고 음질이 떨어지는 문제점이 있다. 하지만 본 논문에서 제시된 기법의 경우(case 4) 일부 프레임에 대해서만 SOLA 알고리즘을 적용하기 때문에 지연도 줄어들면서 음성의 통화품질을 높일 수 있다. 즉, playout 지연시간의 상태에 따라 수신 프레임에 SOLA 알고리즘 적용 여부를 결정하기 때문에 수신된 모든 프레임에 대해서 SOLA 알고리즘을 적용하지는 않는다. Case 3과 4의 첫 번째 프레임의 playout 시점이 case 1의 경우보다 조금 지연된 것은 SOLA 알고리즘 계산시간만큼 추가되었기 때문이다.

수신측에서 수신된 프레임을 손실없이 playout하기 위해서는 그림 5에서와 같이 $s_i = p_i - a_i > 0$ 인 상태가 유지되어야 한다. Adaptive playout buffering 기법에서는 이러한 조건을 만족시키기 위해 이전 패킷들의

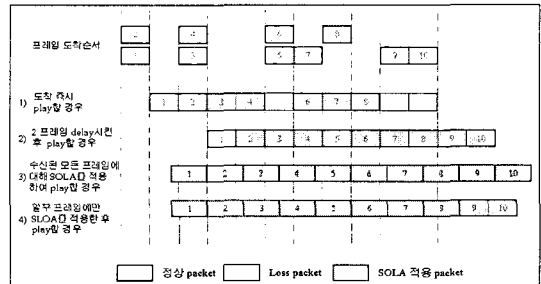


그림 10 SOLA를 이용한 Playout 과정 예

중단간 평균 지연시간과 편차 값을 이용하여 최적의 playout 시간을 결정한 후 해당 시간까지 패킷의 playout시간을 지연시킨다. 이 기법의 경우 각 talkspurt내의 첫번째 패킷에 대해서만 최적의 playout 시간을 찾는 알고리즘을 적용시키고($p_i = t_i + \hat{a}_i + 4\hat{v}_i$), 이후 패킷의 경우에는 송신측에서 생성된 일정한 시간 간격으로 playout 시간($p_j = p_i + t_j - t_i$)을 설정한다. 따라서 첫 번째 패킷의 playout 시간이 결정되고 난 후 talkspurt내의 다른 패킷들의 전송지연시간이 상당히 큰 경우에는 playout되지 못하고 손실패킷으로 처리되는 문제점이 발생할 수 있다. 또한 최적의 playout 시간까지 playout을 지연시키기 때문에 중단간의 지연시간이 증가되어 음성품질을 저하시키는 요인이 될 수 있다.

하지만 본 논문에서 제시된 기법은 각 프레임에 대한

$s_i > 0$ 조건을 만족시키기 위해 수신된 각 프레임의 사람이 인지할 수 없는 범위 내에서 확장시킨다 즉, 수신된 프레임(N)에 변환비율 $\alpha=1.25$ 로 SOLA 알고리즘을 적용하면 수신 프레임의 크기가 $N * 1.25$ 로 증가한다. 예를 들면 송신 프레임의 크기를 40ms로 가정하면 수행될 수신측 프레임 크기는 약 50ms로 증가된다. 따라서 수신측에서 프레임이 playout되는 시점은 SOLA 알고리즘을 적용하지 않을 경우 $p_j = p_i + 40ms$ 로 설정되지만 SOLA 알고리즘을 적용할 경우 $p_j = p_i + 50ms$ 로 적용되어 매 프레임마다 10ms의 playout 지연시간이 발생한다. 이러한 playout 지연시간을 계속 누적시키게 되면 전송지연시간이 커져 playout되지 못하는 프레임도 해당 시간안에 playout을 시킬 수 있다.

또한 Adaptive playout buffering 기법의 경우 최적의 playout 시간 계산을 매 talkspurt내의 첫번째 패킷에 대해서만 수행하고 이후 패킷에 대해서는 송신측에서 전송된 일정한 시간 간격으로 playout 시간을 결정하기 때문에 이후 패킷의 전송지연시간이 클 경우 손실이 발생할 수 있지만 제시된 기법에서는 수신된 매 프레임에 대해 SOLA 알고리즘을 적용하기 때문에 전송지연으로 인한 패킷 손실이 발생할 확률이 적다.

4. 시뮬레이션 및 고찰

본 논문에서는 제시한 알고리즘의 성능을 평가하기 위하여 그림 11과 같은 시뮬레이터를 구성하였다.

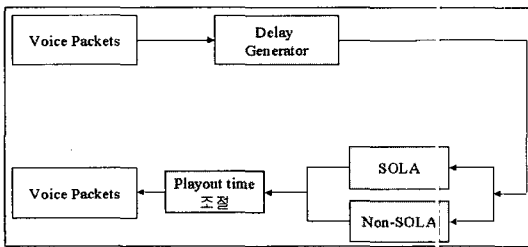


그림 11 시뮬레이터 구성도

Voice Packet에서는 8KHz 8-bit 모노로 샘플링된 프레임을 40ms 간격으로 나누어서 Delay Generator로 전송한다. Delay Generator에서는 수신된 프레임을 데이터로 사용하여 Ping을 수행한다. Ping을 통해 수신된 프레임과 시간정보를 이용하여 일정한 playout 지연시간을 확보할 때까지($Playout_time - (t_i + n_i) > 2 * \max(n_i)$) 각 수신 프레임에 SOLA를 적용시켜서 프레임의 크기를 확장시킨 후 playout한다. playout 지연

시간이 확보되면 정상적인 수신프레임을 playout하다가 playout 지연시간이 부족하면 다시 수신프레임을 확장시켜 playout한다. 각 프레임에 SOLA 알고리즘을 적용할 때 입력 프레임의 크기(N)를 320 바이트, $S_a = N/4$ 로 설정하였다.

본 논문에서는 SOLA 알고리즘을 이용하여 전송지연으로 인한 패킷 손실을 최소화하는 방법으로 기존의 음대체 기법 및 직전 패킷 대체 기법과 음질 비교를 수행하였다. SOLA 알고리즘은 수신측에서 PCM으로 디코드된 패킷에 적용시켜서 수신된 패킷을 확장한다. 청취평가는 비교대상이 되는 음성을 어떤 기법이 사람이 듣기에 더 명료한 소리를 나타내는가를 판단하는 형태로 진행하였다. 30명의 청취자가 실험에 참여하였으며, 청취자 자신이 어떤 음성 데이터가 어떠한 방법으로 복원된 신호인지를 알 수 없도록 음성 데이터 순서를 무작위로 제시하였다.

그림 12는 다양한 전송프레임의 크기에 따른 전송 지연 패킷 손실률을 나타낸 것이다. 전송 프레임의 크기가 적을 경우 각 프레임에 SOLA 알고리즘을 적용하면 발생할 수 있는 playout 지연시간이 그만큼 감소된다. 즉, 20ms 전송 프레임에 $\alpha=1.25$ 인 SOLA 알고리즘을 적용하면 playout 지연시간이 약 5ms가 생성되고, 30ms 전송 프레임에 적용하면 약 7.5ms가 생성된다. 따라서 전송 프레임의 크기가 줄게되면 발생하는 playout 지연시간의 크기도 그만큼 줄게되어 전송지연으로 인한 패킷의 손실 확률이 증가된다. 또한 SOLA를 적용시켜야 되는 프레임의 수도 증가되어 시스템의 오버헤드가 증가한다.

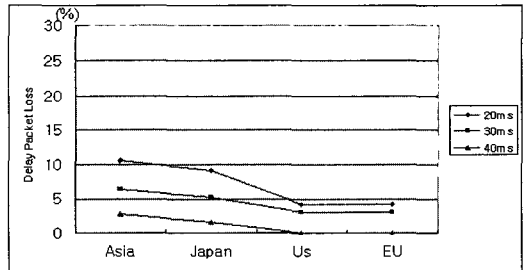


그림 12 프레임 크기에 대한 전송지연 패킷 손실률

그림 13은 각 패킷들의 playout 시점에 대해 SOLA를 적용한 패킷과 적용하지 않은 패킷들간의 playout 시간을 보여주고 있다. 그림 13에서 보는 바와 같이 송신측에서 전송한 패킷은 인터넷을 통해 전송되면서 임

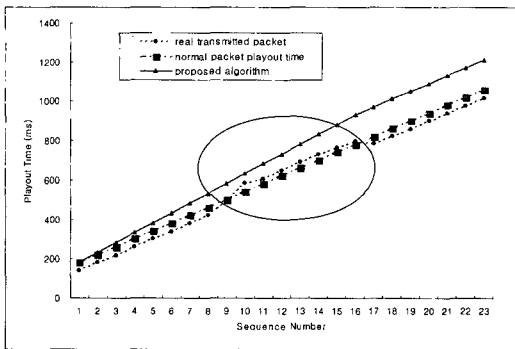


그림 13 SOLA를 적용한 패킷들의 playout 시간

의 지연시간을 겪게 된다. 따라서 수신측에 도달되는 형태를 보면 1번 ~ 9번 패킷, 17번 ~ 23번 패킷은 자신의 playout 시점 이전에 도달하여 제대로 playout을 할 수 있다. 하지만 10번 ~ 16번 패킷은 자신의 playout 시점 이후에 도달하여 playout을 할 수 없어 손실로 간주된다. 하지만 본 논문에서 제시된 기법의 경우 첫 번째 패킷부터 $\alpha=1.25$ 로 SOLA 알고리즘을 적용하여 playout 시점을 늘려 나갔다. 따라서 그림 13에서 보는 바와 같이 정상적인 playout 시점보다는 조금 지연되어 playout됨을 알 수 있다. 하지만 이러한 playout 지연시간을 이용하면 손실로 간주되는 10번 ~ 16번 패킷이 SOLA를 적용한 패킷 playout 시점 이전에 도달한 것으로 나타나기 때문에 정상적으로 playout 할 수 있다.

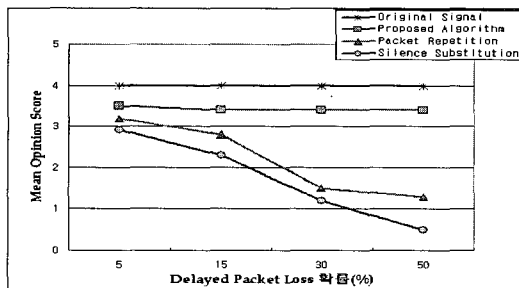


그림 14 지연으로 인한 패킷 손실에 대한 MOS

그림 14와 그림 15는 각 복원 기법으로 복원된 음성 데이터를 청취한 결과를 다양한 전송지연으로 인한 패킷 손실에 대한 MOS와 지연손실 15%이하일 때 MOS 측정 결과를 나타낸 것이다. 그림에서 보는 바와 같이 기존 기법보다 본 논문에서 제시한 기법이 이전 패킷으

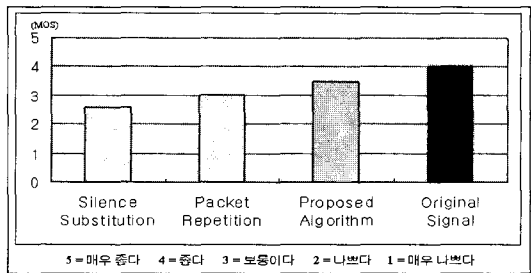


그림 15 지연손실 15%이하일 때 각 기법들의 MOS

로 대체하거나 목음 대체 기법보다 우수한 음질을 나타내고 있음을 알 수 있다. 그림 14에서 지연 패킷 손실 확률은 전송도중에 패킷 손실없이 수신측에 도달했지만 자신의 playout 시간보다 늦게 도착하여 playout되지 못한 경우를 의미한다. 이 경우 목음대체 기법이나 직전 패킷 대체 기법은 전송지연으로 인해 손실로 간주된 패킷을 보상하기 위해 목음이나 직전 패킷으로 대체하기 때문에, 네트워크상에서 체중으로 인한 지연이 발생되면 음질이 현저히 떨어진다. 하지만 본 논문에서 제시된 기법은 첫번째 패킷부터 SOLA를 적용시켜서 playout 시점을 계속 지연시켜 나가기 때문에 전송지연으로 인한 패킷손실을 예방할 있어 음질의 변화는 거의 나타나지 않았다.

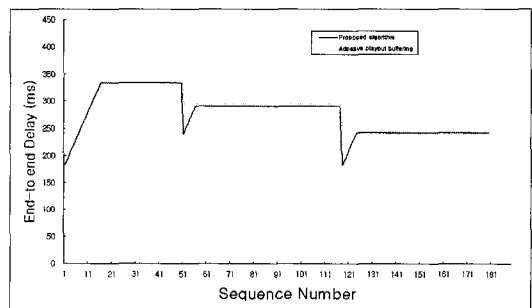


그림 16 Adaptive playout buffering과 제시된 알고리즘과의 중단간 지연 비교

그림 16은 제시된 알고리즘과 adaptive playout buffering 알고리즘간의 중단간 지연을 비교한 것이다. Adaptive playout algorithm의 경우 각 talkspurt내의 첫 번째 패킷의 playout time이 계산되면 n 프레임 딜레이와 마찬가지로 해당 playout time까지 음성패킷의 플레이를 지연시키게 된다. n 프레임 딜레이 기법은 각 talkspurt내의 첫 번째 패킷의 지연시간을 n 프레임만큼

고정된 형태로 지연시키지만 adaptive playout buffering 알고리즘은 각 talkspurt내의 첫 번째 패킷의 playout 시간을 네트워크의 전송지연과 종단간의 지연 편차를 이용하여 가변시킬 수 있는 알고리즘이다. 따라서 네트워크의 전송지연이 클 경우 playout 시간이 함께 증가하고 playout 시점까지는 수신측에서 각 talkspurt내의 첫 번째 음성 패킷을 playout하지 않기 때문에 수신자는 해당 시간만큼 playout 지연을 겪게 된다.

하지만 본 논문에서 제시된 알고리즘은 수신측에서 각 talkspurt내의 첫 번째 패킷부터 SOLA 알고리즘을 적용하여 패킷의 크기를 인간이 인지하지 못하는 수준으로 확장·playout함으로써 수신측에서 각 패킷이 플레이되는 시점을 자연스럽게 지연시킬 수 있도록 하였다. 즉, 수신자는 송신측에서 전송되는 음성을 지연없이 즉시 듣게 되면서, 한편으로는 수신자는 거의 인식하지 못하면서 n 프레임 전송지연이나 adaptive playout buffering algorithm 효과를 얻을 수 있게 된다.

5. 결론

현재 인터넷은 Best-effort형 패킷전달 서비스만을 제공하고 있기 때문에 지연, 패킷손실 및 지터 등을 보장할 수 없다. 인터넷을 이용하는 인터넷폰에서 전송지연에 의한 음성패킷 손실은 음질에 매우 심각한 영향을 야기시킨다. 본 논문에서는 수신측에서 전송지연으로 인한 패킷 손실을 줄이기 위해 수신된 프레임에 SOLA 알고리즘을 적용하여 각 프레임의 크기를 사람이 인지할 수 없는 범위 내에서 확장시켰다. 이것은 각 프레임의 playout 시간을 사람이 인지할 수 없는 범위 내에서 확장시키는 효과를 얻을 수 있기 때문에 이후 프레임에 대한 playout 시간에 여유가 발생한다. 이러한 playout 여유시간을 이용하여 전송지연으로 인한 패킷 손실을 최대한 보상할 수 있게 된다. 또한 모든 프레임에 대해 SOLA 알고리즘을 적용하여 원음에 비해 과다하게 확장되어 종단간의 지연시간이 증가되는 것을 방지하기 위해 playout 여유시간의 양에 따라 각 프레임에 대한 SOLA 알고리즘의 적용 여부를 결정하기 때문에 종단간의 지연도 줄이면서 사람이 인지하는 데는 큰 영향을 주지 않는다. 시뮬레이션 결과 네트워크 상에서 체증으로 인한 전송지연 때문에 발생하는 패킷 손실이 증가할 경우에도 다른 기법에 비해 음질의 변화가 거의 나타나지 않았다.

앞으로는 목음에 대해 변환비율을 유성음과 다르게 적용하여 음질을 보다 더 개선시키는 방식에 대해 계속

연구할 계획이다.

참고 문헌

- [1] David Newman, "VOIP Gateways:Voicing Doubts?," Data Communications, pp. 71~78, Sept. 1999.
- [2] ITU-T Recommendation G.114, "One-way Transmission Time," 1996.
- [3] Thomas J. Kostas, Michael S. Borella, Ikhlak Sidhu, Guido M. Schuster, Jacek Grabiec, and Jerry Mahler, "Real-Time Voice Over Packet-Switched Networks," IEEE Network, Jan/Feb 1998.
- [4] J.C.Bolat, "Characterizing End-to-End Packet Delay and Loss in the Internet," Journal of High-Speed Networks, Vol.2, No.3, pp. 305~323, Dec. 1993.
- [5] Draft ITU-T Recommendation I.380, "Internet Protocol Data Communication Service - IP Packet Transfer and Availability Performance Parameters," 1998.
- [6] ETSI/TIPHON TR 101 329 v2.2.2, "General Aspects of Quality of Service(QoS)," 1999.
- [7] 황원주, "인터넷전화 서비스를 지원하기 위한 Differentiated Service의 성능 평가", 부산대학교 공학석사 학위논문, 2000.
- [8] C.S. Perkins, O.Hodson and V.Hardman, "A Survey of Packet Loss Recovery Techniques for Streaming Audio," IEEE Network Magazine, pp. 40~48, Sept./Oct. 1998.
- [9] R. Ramjee, J. Kurose, D. Towsley, H. Schulzrinne, "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks," IEEE Infocom, June 1994.
- [10] M. R. Portnoff, "Time-scale Modification of Speech Based on Short-time Fourier Analysis," IEEE Trans. Acoustic., Speech, Signal Processing, Vol. ASSP-29, No. 3, pp. 374~390, Jun. 1981.
- [11] T. F. Quateri and R. J. McAulay, "Shape Invariant Time-scale and Pitch Modification of Speech," IEEE Trans. Signal Processing, Vol. 40, No. 3, pp. 497~510, Mar. 1992.
- [12] S.Roucos and A.M.Wilgud, "High Quality Time-Scale Modification for Speech," Proc. ICASSP, pp. 493~496, Apr. 1986.
- [13] E. Moullines and F. Charpentier, "Pitch Synchronous Waveform Processing for Text-to-Speech Synthesis Using Diphones," Speech Communication, Vol. 9(5/6), pp. 453~467, 1990.
- [14] Ryoji Suzuki and Masayuki Misaki, "Time-scale Modification of Speed Signals using Cross-

Correlation Function," IEEE Transactions on Consumer Electronics, Vol. 38, No. 3, pp. 357~363, 1992.

- [15] Van Jacobson, "Congestion avoidance and control," ACM SIGCOMM, pp. 314~329, 1988.



남재현

1989년 2월 부산대학교 컴퓨터공학과(학사). 1992년 2월 부산대학교 컴퓨터공학과(석사). 1996년 3월 ~ 현재 부산대학교 컴퓨터공학과 박사과정. 1993년 3월 ~ 현재 동주대학 컴퓨터정보통신계열 조교수. 관심분야는 VoIP, 개인이동통신,

TCP/IP 프로토콜 등



이정태

1976년 부산대학교 공과대학 전자공학과(학사). 1983년 서울대학교 공과대학 컴퓨터공학과(석사). 1989년 서울대학교 공과대학 컴퓨터공학과(공학박사). 1977년 3월 ~ 1977년 12월 한국과학기술원 연구원. 1977년 12월 ~ 1985년 2월 한국

전자통신연구소 선임연구원. 1985년 3월 ~ 1988년 2월 동아대학교 공과대학 조교수. 1992년 8월 ~ 1993년 7월 일본 NTT 연구소 초빙연구원. 1988년 3월 ~ 현재 부산대학교 컴퓨터공학과 교수. 관심분야는 고속 트랜스포트 프로토콜, 인터넷 서비스, ATM 프로토콜, 개인이동통신, VoIP