
인터넷전화에서 지터보상을 위한 Frame Extension for Adaptive Playout Time(FEAPT) 알고리즘

남재현*

FEAPT Algorithm to compensate Jitter in Internet Phone

Jae-Hyun Nam*

요약

인터넷 전화 서비스는 저렴한 가격과 타 서비스와 통합 및 가치부가면에서 기존의 전화에 비해 많은 장점을 가지고 있으나, 상대적으로 낮은 음질로 인하여 사용자의 요구를 만족시키지 못하고 있다. 이것은 인터넷이 best-effort형 패킷 전달 서비스만을 제공하고 있기 때문에 지연, 패킷 손실 및 지터 등을 보장할 수 없기 때문이다. 본 논문에서는 인터넷 전화에서 전송지연으로 인한 패킷손실을 감소시키기 위해 수신 패킷을 사람이 인지 가능한 수준까지 확장하여 playout 시킬 수 있는 FEAPT 알고리즘을 제시한다.

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 FEAPT algorithm in Internet phone. In the FEAPT algorithm, the receiver expands the received packet under resonable threshold, and hence compensates the QoS of speech.

키워드

Internet Phone, Packet loss, Delay, FEAPT Algorithm, SOLA Algorithm

I. 서 론

컴퓨터 기술 및 광통신 및 라우터 기술을 비롯한 통신 기술의 발전에 급속한 발전은 인터넷 사용인구의 급증과 실시간 트래픽을 수용하는 방향으로 주 사용 서비스의 변화를 가져왔다. 이러한 변화는 네트워크 구조에도 많은 영향을 끼쳐, 공중망의 중심이 회선교환망에서 패킷교환망으로 바뀌고 있고, 결국 PSTN이 IP 네트워크에 통합되는 결과를 가져 올 전망이다[HWANG99]. 이러한

통합은 인터넷과 같은 IP 네트워크를 통해 음성, 비디오, 팩스 등 다양한 서비스를 전송할 수 있음을 의미한다. 이와 같이 인터넷에는 새로운 서비스가 끊임없이 추가되고 있는데, 그 중 대표적인 것이 인터넷을 통하여 전화서비스를 제공하는 인터넷 전화 서비스이다.

인터넷 전화란 인터넷을 이용하여 전화 혹은 음성 서비스를 제공하는 기술이다. 이러한 서비스는 저렴한 가격과, 타 서비스와 통합 및 가치부가면에서는 기존의 전화에 비해 많은 장점을 가지고

*신라대학교 컴퓨터정보공학부 전임강사

접수일자 : 2003. 9. 18

있으나, 상대적으로 낮은 음질로 인하여 사용자의 요구를 만족시키지 못하고 있다. 이는 현재 인터넷에서는 Best-effort형 패킷 전달 서비스만을 제공하고 있기 때문에 네트워크상에서의 체증이 심하게 발생하여 전송지연이나 패킷 손실이 많을 경우 음질이 크게 떨어지기 때문이다[1]. 따라서 인터넷을 통하여 음성을 전달할 경우에는 지연(delay), 패킷 손실(packet loss) 및 지터(jitter) 등의 통화 품질 열화 요인들에 대한 문제점이 해결되어야만 사용자가 만족할 수 있는 음성품질을 얻을 수 있다.

인터넷 전화의 품질을 저하시키는 요인 중의 하나인 패킷 손실 문제를 해결하기 위한 연구들은 크게 전송도중 발생된 패킷 손실을 보상하기 위해 송·수신측에서 손실된 패킷을 보상하는 패킷 손실 보상기법[2]과 전송지연으로 인해 수신측에 제시간에 도달하지 못해 손실로 간주되는 패킷의 수를 줄이기 위한 playout 버퍼링 기법 등으로 분류할 수 있다. playout 버퍼링 기법 중 가장 많이 사용되고 있는 기법은 adaptive playout buffering 알고리즘[3]이 있다.

하지만 패킷 손실 보상 기법은 연속적인 패킷 손실이 발생할 경우 손실 패킷을 보상하더라도 음질이 상당히 저하되고, playout 버퍼링 기법의 경우에는 네트워크 전송지연시간을 정확히 측정하기 어렵고 종단간의 지연이 증가함에 따라 음질이 저하된다. 또한 playout buffering 알고리즘의 경우 수신측에서 각 패킷의 playout 시점을 정확히 예측하려면 송수신측간 시스템 클럭의 동기를 유지해야 하고, playout 시점을 계산하기 위해 각 수신 패킷마다 종단간의 평균지연과 지연 편차를 계산을 계속 수행한다. 이는 오버헤더가 많이 소요되기 때문에 실시간 처리에 있어 부담을 주게 된다.

본 논문에서는 인터넷 전화의 품질을 저하시키는 요인 중의 하나인 전송 지연으로 인한 패킷 손실 문제를 해결하기 위해 FEAPT(Frame Extension for Adaptive Playout Time) 기법을 제안하고 그 성능을 분석하였다. FEAPT 알고리즘은 수신측에서 수신된 음성 프레임에 SOLA 알고리즘을 적용하여 시간축으로 확장한 후 playout 함으로써 발생되는 시간 여유로 전송지연으로 인해 제

시간에 playout 되지 못하는 프레임의 수를 줄이는 기법이다. 본 논문에서 제시된 기법의 성능을 분석하기 위해 PC-to-PC 환경의 인터넷 전화를 구현하여 기존의 playout buffering 알고리즘과 성능 비교를 수행하였다.

II. 인터넷 트래픽 특성 분석

본 논문에서는 인터넷 전화의 음성 품질에 영향을 주는 요소들을 분석하기 위해 Ping을 이용하여 패킷 전송지연, 손실, 지터 등을 측정하였다. 측정한 구간은 국내 21곳, 북미 21곳, 유럽 20곳, 아시아/태평양 19곳, 총 4개 그룹에 81개의 사이트를 대상으로 측정하였다.

1. 지연시간

지연시간은 송신측에서 수신측으로 하나의 패킷을 전송하는데 소요되는 시간이다. 송신측에서 전송된 각 패킷은 경로 중간의 여러 라우터를 거쳐 최종적으로 수신측에 도달한다. 인터넷에서의 종단간의 지연시간(end-to-end delay time)은 단방향으로 5ms에서 500ms까지 다양하게 분포되어 있다[4].

이러한 지연시간이 너무 크면 인터넷전화에서 상대방과 통화를 할 때 원활한 대화가 어려워진다. ITU-T에서는 종단간의 전화통화에서 단방향 지연시간을 400ms 이내로 권고하고 있다[5]. 그러나 [6]에서는 단방향 지연시간이 100ms 이내로 되어야 인터넷전화에서 toll급의 통화품질을 얻을 수 있다는 연구 결과를 제시하고 있다.

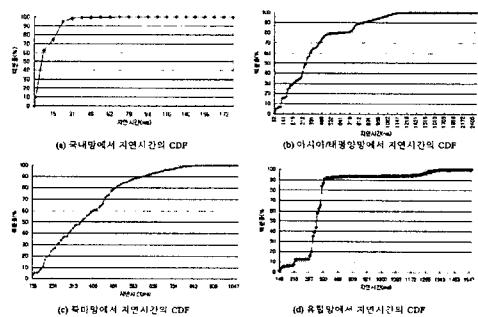


그림 1. 각 지역망에서 측정된 지연시간
Fig. 1. CDF of round trip delay in Internet

그림 1에는 국내망, 아시아/태평양, 북미망, 그리고 유럽망을 대상으로 측정한 지연시간을 누적분포 함수(Cumulative Distributed Function, CDF)를 이용하여 도시하였다.

그림 1을 살펴보면, 국내망의 경우 15ms에서 172ms까지 다양하게 분포되어 있지만 전체 패킷 중 90%가 16ms 이내로 분포되어 있다. 하지만 아시아/태평양망의 경우 62ms에서 2406ms까지, 북미망의 경우 156ms에서 1407ms까지, 유럽망의 경우 140ms에서 1547ms까지 전송지연이 상당히 크게 분포되어 있다. 이들 지역의 전체 패킷 중 90%는 아시아/태평양망의 경우에는 844ms이내에, 북미망의 경우에는 609ms이내에, 그리고 유럽망의 경우에는 391ms이내로 분포되어 있다.

2. 지연 편차

지터 또는 지연 편차(delay variation)는 전송지연시간의 편차를 의미하는데, 이 값이 너무 크면 수신측에서 음성이 일정한 간격으로 재생되지 못하고, 일그러지는 현상이 발생하므로 패킷손실과 마찬가지로 음성품질을 저하시킨다. 이러한 지터는 75ms 이내이어야 만족할 만한 수준의 품질을 얻을 수 있다.[7]. 본 논문에서는 지터의 특성을 조사하기 위해 전송시간과 수신측에 각 패킷이 도달되는 시간 간격(interarrival time)간의 관계를 측정하였다.

그림 2에는 국내망, 아시아/태평양망, 북미망, 그리고 유럽망을 대상으로 측정한 패킷 도착 시간 간격을 누적분포함수를 이용하여 도시하였다.

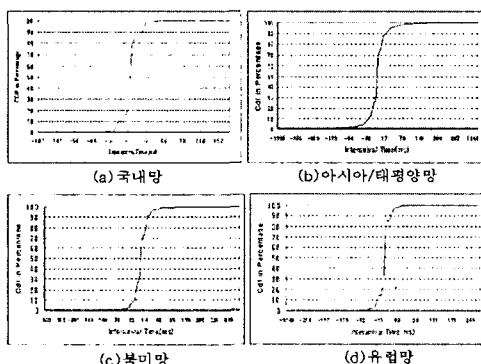


그림 2 각 지역망에서의 패킷 도착 시간 간격
Fig 2. CDF of Internet interarrival times

그림 2에서 보는 바와 같이 국내망, 아시아/태평양망, 북미망, 유럽망의 경우에는 전체 패킷 중 90%가 15m, 30m, 16ms, 15ms이내로 나타나지만 나머지 10%는 187ms, 1422ms, 500ms, 282ms 등 다양한 분포를 나타내고 있다. 이와 같이 현재의 인터넷에서는 다양한 전송지연으로 패킷이 수신측에 일정한 시간 간격으로 전송될 수 없다. 따라서 수신측에서는 지터 문제를 보완하기 위해 수신측에서 수신 패킷을 일정시간 지연시켜서 playout을 수행한다.

3. 패킷 손실

패킷손실은 라우터에서의 폭주 또는 음성과 같은 실시간 응용에서 패킷이 수신측에서 playout되는 시점보다 늦게 도착할 경우 발생되는데 패킷손실이 증가되면 통화품질이 저하된다. 음성통신에서 원활한 품질을 유지하기 위해서는 패킷 손실률은 약 5%이내로 되어야 한다[8]. 일반적으로 인터넷에서의 패킷 손실률은 20%까지 다양하게 분포되어 있다[4].

표 1은 각 그룹별로 측정한 지연시간과 패킷손실률을 정리하여 보여주고 있다. 지연시간은 CDF의 95%, 90%, 85%지점에서의 지연시간으로 나타내었다.

표 1 인터넷에서의 지연시간, 패킷손실률
Table 1. Delay and packet loss rates in Internet

	지연시간 (ms)			패킷손실 (%)
	85%	90%	95%	
국내망	16	16	31	0.03
북미망	532	609	688	8.36
아시아/태평양망	766	844	1015	12.2
유럽망	390	391	1219	13.4

표 1에 제시한 지연시간 CDF가 90%인 지점 ($D_{90\%}$)과 패킷손실의 측정치를 다시 그림 4에 나타내어 보았다. 표 1에 제시한 지연시간은 양방향 지연시간이므로 $D_{90\%}/2$ 의 값을 지연시간 CDF가 90%인 지점의 단방향 지연시간이라고 가정하였다[9].

지연시간과 패킷손실에 따른 음성품질의 실측 결과를 나타낸 그림 3을 살펴보면, 국내망의 경우

인터넷전화는 toll급의 통화품질을 얻을 수 있지만 국제망의 경우에는 아직 품질이 만족스럽지 못함을 알 수 있다.

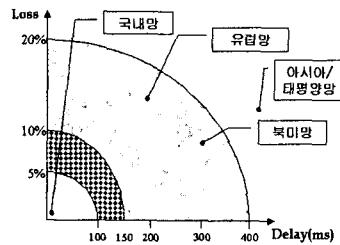


그림 3. 지연시간과 패킷손실에 따른 음성품질 실측 결과
Fig 3. QoS mapping for Internet delay and loss

III. 전송지연으로 인한 패킷 손실 보상을 위한 FEAPT 알고리즘

본 논문에서 제시된 FEAPT 알고리즘은 수신측에서 전송지연으로 인한 패킷 손실을 줄이기 위해 SOLA(Synchronized Overlap-Added) 알고리즘을 이용하였다.

1. SOLA 알고리즘

SOLA 알고리즘은 음성 신호의 시간축 변환을 위해 프레임 단위의 신호들을 정규화된 상관 합수(normalized cross-correlation function)를 이용하여 축소 또는 확장한 후, 변환된 프레임간의 동기를 맞추고 중첩 가산(overlap and add)하는 방법을 사용하는 기법이다[10]. 즉, 오디오 신호의 재생 속도를 변화시켜 원래 신호보다 빠르게 혹은 느리게 음을 합성하여 음성신호가 가지는 중요한 스펙트럼 정보는 그대로 유지하면서 단지 발음 속도만을 변환시키는 기법으로, 실시간 처리가 가능하면서 비교적 고음질의 변환 신호를 얻을 수 있다. SOLA 알고리즘은 그림 4와 같이 수행된다.

음성 신호 $x(n)$ 을 변환비율 a 에 의해 변환시켜서 $y(n)$ 이라는 합성 신호를 얻는다고 하자. 음성 신호 $x(n)$ 에서 매 분석구간 S_a 마다 N 개의 샘플로 구성된 프레임들을 가지고 매 S_s 마다 합성신호 $y(n)$ 을 합성하는데 사용한다면, 분석 구간 S_a 와

합성구간 S_s 사이에는 $S_s = aS_a$ 라는 관계가 성립 한다. 따라서 a 가 1보다 클 경우 확장의 의미를 가지고 1보다 적을 경우에는 축소한다는 의미를 지닌다.

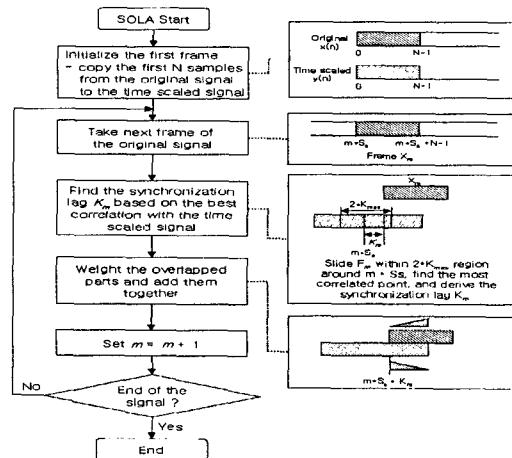


그림 4. SOLA 알고리즘

Fig 4. SOLA algorithm

2. FEAPT 알고리즘

FEAPT 기법은 수신측에서 수신된 패킷 각각에 SOLA 알고리즘을 적용하여 수신 패킷의 크기를 사람이 인지할 수 있는 범위까지 확장시키는 기법이다. 각 수신 패킷의 크기를 원음에 비해 변환비율 a 만큼 증가시켰기 때문에 변환된 수신 패킷의 playout 시간은 원 패킷에 비해 변환비율 a 만큼 여유를 지니게 된다. 이러한 여유시간을 계속 누적시켜 나가게 되면 수신측에서 늦게 도착하여 playout되지 못하는 패킷의 수를 줄일 수 있게 된다. 따라서 FEAPT 기법은 별도의 버퍼링 기법을 사용하지 않고도 버퍼링 효과를 얻을 수 있는 장점을 지닌다.

기존의 SOLA 알고리즘의 경우 전체 입력 음성 신호를 수신한 후 음성 신호를 S_a 간격으로 분할한 후 이를 다시 S_s 간격으로 재배치·합성하여 음성신호를 늘리거나 줄이는 방법을 사용했다. 하지만 FEAPT에서는 SOLA를 전체 음성신호에 적용하지 않고 수신된 각 패킷에 대해 수신측의 playout 여유시간의 상태에 따라 SOLA를 적용하였다. 따라서 수신 패킷의 직전이나 직후 패킷의

보존없이 수신 패킷 자체를 늘리기 때문에 수신측 버퍼의 양도 줄일 수 있다. FEAPT 알고리즘은 그림 5와 같다.

FEAPT 알고리즘은 먼저 수신측에 수신된 프레임이 talkspurt내의 첫번째 프레임인지를 조사한다. 첫번째 프레임이면 수신 프레임에 SOLA 알고리즘을 적용하여 프레임의 크기를 확장한 후 playout을 수행한다.

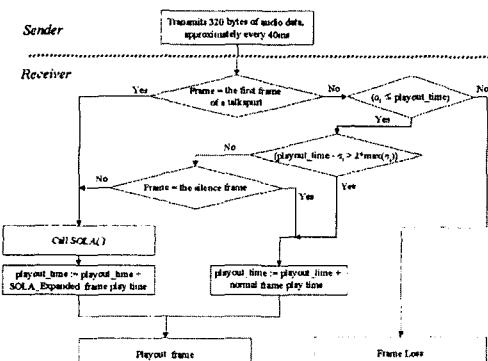


그림 5. SOLA 기법을 이용한 FEAPT 알고리즘

Fig 5. FEAPT algorithm using SOLA

첫번째 프레임이 아닐 경우 수신 프레임의 도착시간(ai)이 자신의 playout 시점 이전에 도착했는지 조사한다. 자신의 playout 시점 이후에 도달한 프레임은 손실로 간주되고 그렇지 않을 경우 playout 지연시간을 조사한 후 여유시간이 충분히 있는지 조사한다. 여유시간이 부족할 경우 수신 프레임이 묵음인지를 조사하여 묵음이 아닌 프레임에 대해서만 SOLA 알고리즘을 적용한 후 확장된 프레임의 playout을 시작한다. 묵음 프레임이나 여유시간이 충분할 경우 정상적인 프레임을 playout한다.

수신측에 수신된 수신 프레임 x를 변환 비율 α 를 적용하여 시간축으로 확장 또는 축소 변환된 프레임 y를 얻을 경우 수행되는 과정은 그림 6과 같다.

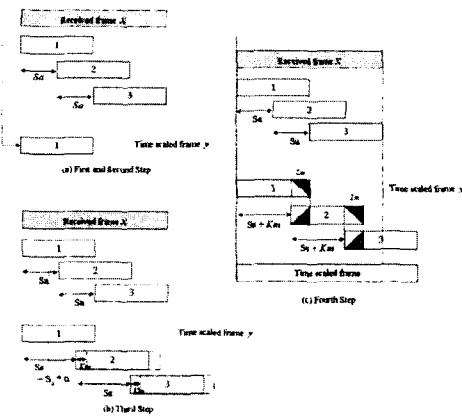


그림 6. SOLA를 이용한 FEAPT 기법 수행 과정

Fig 6. FEAPT operation step

송신측에서 샘플링된 프레임을 전송하면 수신측에서는 수신프레임에 SOLA 알고리즘을 적용하여 수신프레임의 크기를 사람이 인지할 수 있을 정도까지 확장한다. 예를 들면 송신 프레임의 크기를 40ms로 가정하고 변환비율 $\alpha=1.3$ 으로 SOLA를 적용하면, 수행될 수신측 프레임 크기는 약 52ms로 증가된다. 따라서 수신측에서 프레임이 playout되는 시점은 SOLA를 적용하지 않을 경우 $p_j = p_i + 40ms$ 로 설정되지만 SOLA를 적용할 경우 $p_j = p_i + 52ms$ 로 적용되어 매 프레임마다 12ms의 여유시간이 발생한다. 이러한 여유시간을 계속 축적하여 전송지연으로 인한 프레임 손실을 감소시키는데 사용한다.

그림 7과 그림 8는 각 프레임이 송신측에서 전송된 후 수신측에 playout되는 과정을 나타내었다.

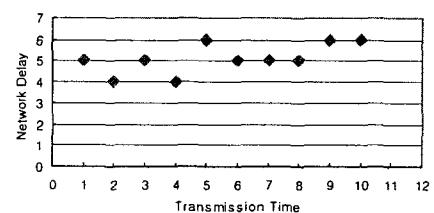


그림 7. 프레임 i의 전송시간 및 네트워크 지연시간

Fig 7. Transmission time and Network delay in frame i

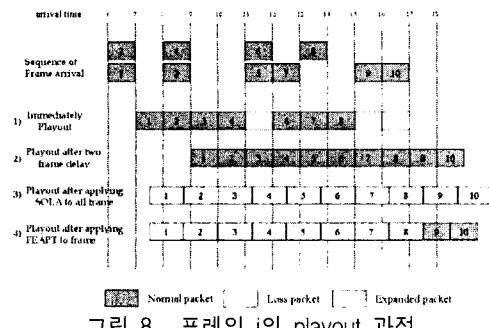


그림 8. 프레임 i의 playout 과정
Fig 8. Example of playout mechanisms in frame i

그림 7은 송신측에서 전송되는 각 프레임이 수신측에 도달할 때까지 겪는 지연시간을 나타낸 것이다. 예를 들면 송신측에서 "1"이라는 시간대에 전송된 1번 프레임은 네트워크상에서 "5"라는 시간대에 도달하게 된다. 수신측에 도착한 시간대와 playout 되는 시간대는 그림 8에 나타나 있다.

그림 8의 1)의 경우처럼 수신측에 도착 즉시 playout할 경우 5번, 9번, 10번 프레임은 자신의 playout 시간대 이후에 도달했기 때문에 손실로 처리되지만, 2)의 경우처럼 2 프레임만큼 지연시킨 후 playout할 경우 모든 프레임이 자신의 playout 시간대 이내로 도달되었기 때문에 손실되는 프레임은 없다. 하지만 2)의 경우는 2 프레임 크기만큼 종단간에 지연이 발생된다. 3)의 경우처럼 수신측에 도착하는 모든 프레임에 대해 SOLA를 적용할 경우 모든 프레임이 자신의 playout 시간대 이내로 도달되었기 때문에 손실되는 프레임은 없지만 playout되는 전체 프레임의 크기가 원음에 비해 과다하게 확장되어 종단간의 지연시간이 증가되고 확장된 프레임 수가 많기 때문에 원음에 비해 음질이 다소 떨어지는 문제점이 있다. 하지만 4)의 경우와 같이 FEAPT 기법을 적용할 경우 원음에 비해 과다하게 확장되는 것을 방지하기 위해 여유시간의 상태(PlayOutTime ReceiverTime >= 2 * max(ni))에 따라 프레임에 대한 SOLA 적용 여부를 결정하기 때문에 playout 지연도 줄이면서 음성의 통화품질을 높일 수 있다. 3)과 4)의 경우 첫 번째 프레임의 playout 시점이 1)의 경우보다 조금 지연된 것은 SOLA 알고리즘의 계산 시간만큼 추가되었기 때문이다.

IV. 성능 분석

1. 구현 환경

본 논문에서 제시한 알고리즘의 성능을 평가하기 위한 측정 환경은 그림 9와 같다.

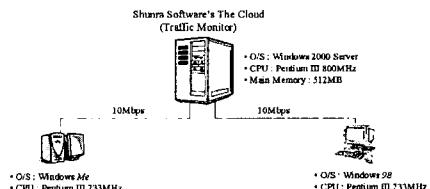


그림 9. 성능 측정 환경
Fig 9 Empirical environment

송신측에서 8KHz 16-bit로 샘플링된 프레임을 40ms 간격으로 나누어서 Cloud 트래픽 모니터로 전송한다. Cloud 트래픽 모니터는 Shunra Software Ltd.에서 개발한 WAN 시뮬레이터로써 다양한 전송지연과 패킷 손실을 시뮬레이팅할 수 있도록 해준다. Cloud WAN 시뮬레이터 환경은 III장에서 조사된 국내망, 북미망, 아시아/태평양망, 유럽망의 전송특성을 기초로 하여 표 2과 같이 설정하였다. 지연시간은 각 지역의 지연시간 분포에서 90%되는 지점으로 설정하였다. 하지만 II장의 표 1에서의 지연시간은 양방향 지연시간이기 때문에 본 논문에서 수행한 WAN 시뮬레이션 환경에서는 D90%/2의 값(지연시간 CDF가 90%인 지점 / 2)을 단방향 지연시간이라고 가정하여 사용하였다.

표 2 시뮬레이션 환경에서의 지연시간과 패킷손실률
Table 2. Delay time and Packet loss rate in each empirical environment

	지연시간(ms)	패킷손실(%)
국내망	20	0.03
북미망	300	8.36
아시아/태평양망	420	12.2
유럽망	200	13.4

수신측은 수신 프레임에 FEAPT를 적용시켜서 프레임의 크기를 확장시켜 나간다. 각 프레임에 SOLA 알고리즘을 적용할 때 입력 프레임의 크기

(N)를 320 바이트, $Sa = N/4$ 로 설정하였다. 이후 수신 프레임에 대한 SOLA 적용 여부를 여유시간의 상태에 따라 결정한 후 프레임의 playout 시간을 조절한다.

```

Private Sub UDPocket_DataArrival(ByVal BytesTotal As Long)
    ReceiverTime = CallLocalTime()
    n_i = ReceiverTime - Sender time 'Propagation delay
    Call GetData(wavedata, vbByte + vbArray, waveChunkSize)
    If Talkspur = True Then 'Start TalkSpur
        If NOT Check_Silence(wavedata) Then 'Check silence
            Call SOLA(wavedata, Alpha) 'Expanded frame
        End If
    Else 'Non First frame
        If ReceiverTime <= PlayOutTime Then
            Diff = PlayOutTime - ReceiverTime
            If Diff <= 2 * max(n_i) Then
                If NOT Check_Silence(wavedata) Then 'Check silence
                    Call SOLA(wavedata, Alpha) 'Expanded frame
                End If
            Else
                Packet_Loss = Packet_Loss + 1 'Packet loss
            End If
        End If
        Call SaveStreamBuffer(Index, wavedata) 'Save wave data to buffer
        Call AddStreamToQueue(Index) 'Queue current stream for playback
    End Sub

```

그림 10. 프레임 수신 모듈
Fig 10. The module of the frame receiver

본 논문에서는 FEAPT 알고리즘의 성능을 분석하기 위해 PC-to-PC 환경의 인터넷 전화 모듈을 구현하였다. 구현된 모듈은 크게 음성을 샘플링하여 전송하는 송신부와 전송된 프레임을 수신한 후 FEAPT 알고리즘을 적용하여 지터를 보상하는 수신부로 구성되어 있다. 그림 10은 수신부에서 동작하는 FEAPT 알고리즘을 보여주고 있다.

2. 성능평가

2.1 패킷 손실률 비교

그림 11은 표2 환경으로 시뮬레이션을 수행했을 때 수신측에 발생되는 패킷 손실율을 나타낸 것이다. 패킷 손실율은 송신측에서 수신측으로 전송되는 도중에 발생한 손실과 전송지연으로 인해 수신측에서 발생된 패킷 손실을 합한 결과이다. 그림 11에서 Non_PBA는 Playout 버퍼링 알고리즘을 적용하지 않는 기법을 의미하고, Adaptive_PBA는 [4]에서 제시된 Adaptive playout buffering 알고리즘을 의미한다.

그림 11을 살펴보면 국내망 환경에서는 전송지연으로 인한 패킷 손실율이 낮아 세 기법 모두 비슷한 패킷 손실율을 보이지만, 국외망인 경우 playout 버퍼링 기법을 사용하는 경우와 사용하지 않는 경우는 상당한 차이를 보이고 있다. 이는 국

외망의 경우 각 패킷들이 경험하는 지터들의 분포가 표장에서 보는 바와 같이 다양하게 분포되어 있다는 것을 의미한다. 따라서 이러한 지터를 보상하기 위해 수신측에 버퍼링 기법을 이용한다면 전송지연으로 인한 패킷의 손실을 상당히 줄일 수 있다.

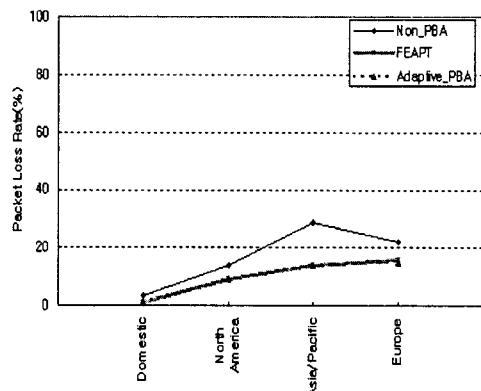


그림 11. 각 시뮬레이션 환경에서의 패킷 손실율
Fig 11. Packet loss rate in each simulation environment

그러나 FEAPT나 Adaptive_PBA와 같이 수신측에서 playout 버퍼링 기법을 사용하는 경우 수신측에서의 패킷 손실율은 거의 차이가 발생하지 않는다. 이는 FEAPT 기법의 경우 수신측에서 수신된 패킷의 크기를 계속 확장함으로써 별도의 버퍼링 알고리즘을 사용하지 않고도 Adaptive_PBA와 동일한 효과를 얻을 수 있기 때문이다.

2.2 playout 지연시간 비교

그림 12는 각 지역 시뮬레이션 환경에서 Adaptive Playout Buffering 알고리즘과 FEAPT의 성능을 비교한 것이다. 성능 비교 인자는 playout 지연시간을 누적분포함수로 나타내었다. 여기서 playout 지연시간은 수신측에서 패킷을 수신한 후 해당 패킷을 playout할 때까지의 시간을 의미한다.

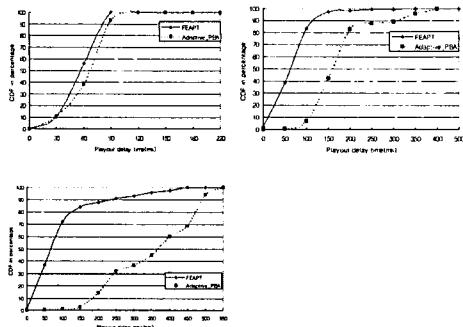


그림 12. 각 시뮬레이션 환경에서의 playout 지연시간
Fig 12. Playout delay time in each simulation environment

국내망의 경우 전체 패킷 중 90%가 Adaptive PBA 인 경우 90ms를 나타내고 FEAPT의 경우 75ms를 나타낸다. 이는 국내망의 경우 전송지연 시간이 크지 않기 때문에 두 기법 간의 playout 지연시간의 차이가 크게 나타나지 않는다.

북미망 시뮬레이션 환경에서는 전체 패킷 중 90%인 지점의 경우, FEAPT 알고리즘은 약 110ms를 나타내고 Adaptive PBA는 약 200ms를 나타낸다. 국내망 환경보다는 많은 playout 지연시간을 나타내는 원인은 Adaptive PBA와 FEAPT 모두 playout 지연시간이 전송시간에 따라 달라지기 때문이다.

아시아/태평양망 시뮬레이션 환경에서 Adaptive PBA의 평균 playout 지연시간은 약 347.72ms로 나타나지만, FEAPT 기법의 경우 약 97.98ms로 나타났다. 아시아/태평양망의 playout 지연시간을 보면, 전체 패킷 중 90%가 FEAPT인 경우 약 220ms를 나타내고 Adaptive PBA의 경우 약 480ms를 나타낸다. 다른 환경보다 많은 playout 지연시간을 나타내는 원인은 Adaptive PBA와 FEAPT 모두 playout 지연시간을 계산할 때 전송시간의 영향을 크게 받기 때문이다. 따라서 FEAPT 알고리즘에서 playout 지연시간이 크지는 문제를 해결하기 위해 다양한 여유시간 베틀을 고려해야 할 것이다.

유럽망 시뮬레이션 환경에서 Adaptive PBA의 평균 playout 지연시간은 약 199.26ms로 나타나지

만, FEAPT 기법의 경우 약 89.9ms로 나타났다. 그림에서 보는바와 같이 전체 패킷 중 90%가 FEAPT인 경우 약 170ms를 나타내고 Adaptive PBA의 경우 약 310ms를 나타낸다.

V. 결 론

본 논문에서는 인터넷의 트래픽 특성을 분석하여, 인터넷을 통해 음성 통신을 수행할 경우 수신측에서 전송지연으로 인한 패킷 손실을 줄일 수 있는FEAPT 기법을 제시하고 성능을 분석하였다. FEAPT 기법 알고리즘은 전송지연으로 인해 자신의 playout 시점 이후에 도달한 각 프레임의 playout시점은 이전 프레임들의 증가된 playout 시간으로 보상할 수 있다. 또한 수신 프레임의 확장이 끝나면 확장된 프레임을 즉시 playout 시키기 때문에 기존의 playout 베틀링 알고리즘보다 초기 베틀링 시간이 상당히 단축시킨다.

제시된 FEAPT 알고리즘의 성능을 평가하기 위해 각 시뮬레이션 환경에서 패킷 손실과 playout 지연시간 측정을 수행했다. 수행 결과 FEAPT 기법은 수신측에서 전송지연으로 인해 자신의 playout 시점 이후에 도달하여 패킷 손실로 간주되는 확률을 약 2% 이내로 줄일 수 있었고, playout 지연시간 역시 기존의 playout 베틀링 알고리즘에 비해 크게 단축시켰다.

하지만 FEAPT 역시 playout 지연시간을 계산할 때 전송시간의 영향을 받기 때문에 playout 지연시간이 크지는 문제를 해결하기 위해 다양한 여유시간 베틀을 고려해야 할 것이다. 앞으로는 이러한 playout 지연시간을 보다 더 줄이기 위해 네트워크 트래픽 상황에 따라 변환비율을 가변적으로 적용하는 방법과 전송도중 발생되는 패킷 손실을 보상하기 위해 기존의 패킷 손실 보상 기법과 혼합하여 사용하는 방식에 대해 계속 연구할 계획이다.

참고 문헌

- [1] David Newman, "VOIP Gateways: Voicing Doubts?", Data Communications, pp. 71~78, Sept. 1999
- [2] C.S. Perkins, O.Hodson and V.Hardman, "A Survey of Packet Loss Recovery Techniques for Streaming Audio", IEE Network Magazine, pp. 40~48, Sept./Oct. 19
- [3] R. Ramjee, J. Kurose, D. Towsley, H. Schulzrinne, "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks", IEEE Infocom, June 1994
- [4] J.C.Bolot, "End-to-End Packet Delay and Loss Behavior in the Internet", SIGCOMM Symposium on Communications Architectures and Protocols, pp.289~298, Sept. 1993
- [5] ITU-T Recommendation G.114, "One-way Transmission Time", 19
- [6] T.J. Kostas, M. S. Borella, I. Sidhu, G. M. Schuster, J. Grabiec, and J. Mahler, "Real-Time Voice Over Packet-Switched Networks", IEEE Network, Jan/Feb 1998
- [9] ETSI. Recommendation GSM 6.11, "Substitution and muting of lost frames for full rate speech channels", 1992
- [8] N. S. Jayant, "Effects of packet losses in waveformcoded speech," in Int. Conference on Computers and Communications, pp. 275~280, Oct. 1980
- [9] 황원주, "인터넷전화 서비스를 지원하기 위한 Differentiated Service의 성능 평가", 공학석사 학위논문, 부산대학교, 2000
- [10] S.Roucos and A.M.Wilgud, "High Quality Time-Scale Modification for Speech", Proc. ICASSP, pp. 493~496, Apr. 1986

저자 소개



남재현(Jae-Hyun Nam)

1989년 부산대학교 컴퓨터공학과
(공학사)
1992년 부산대학교 컴퓨터공학과
(석사)

2002년 부산대학교 컴퓨터공학과 (박사)
1993년 3월 ~2002년 10월 동주대학 네트워크 전자
계열 조교수
2002년 11월 ~현재 신라대학교 컴퓨터정보공학부
전임강사
※관심분야 : 컴퓨터네트워크, VoIP