

인터넷 전화 음질 개선을 위한 지능적인 버퍼관리 방식

류태욱*,강성호*,이정훈*,임중규*,이현관**,엄기환*

*동국대학교, **호남대학교

Intelligent buffer Management Method for QoS of Internet Telephony

Tae-uk Ryu*,Seong-ho Kang*,Jong-hun Lee*, Joong-gyu Lim*,Hyun-Kwan Lee**,

Ki Whan Eom*

*Dongguk Univ. **Honam Univ

E-mail : ryurim@dgu.ac.kr

ABSTRACT

This thesis introduces a buffer management method that could be used to provide better sound quality for Internet phone terminals. Proposed method used the fuzzy system for dynamically adjust the number of jitterThis method actively responds to both the compression algorithms that are used by the terminals, as well as to the received data to provide an improvement in sound quality. In order to confirm the validity of the suggested algorithm, comparisons of the performance have been made between the existing buffer management method and this intelligent method in various network settings.

(GSTN: General switched telephony network)보다 음질이 좋지 못하게 될 수 있다[1][2]. 이에 음질향상을 위한 QoS를 위하여 종단에서 종단까지 많은 연구가 이루어져 왔으나 단말기입장의 연구는 현실성과는 다소 떨어진 입장이 많았다. 실제 다른 연구들에서 고려된 전송 지연 시간의 계산과 음성 목음압축의 사용은 추가적인 정보와 특정 잡음의 재생에 의한 음질의 저하를 유발하게 된다. 이에 본 논문에서는 인터넷전화에서 보다 양질의 음성을 제공하기 위하여 압축알고리즘의 종류와 수신된 패킷정보만을 고려하여 네트워크의 변화에 적응적으로 반응하는 단말기 입장의 지능적인 버퍼관리 방식을 제안한다. 제안한 방식은 퍼지 시스템을 이용하여 자동적으로 버퍼의 수를 결정한다. 제안한 지능적인 방식의 유용성을 확인하기 위하여 다양한 네트워크 상황에서 기존의 버퍼관리방식과 네트워크 상황판단의 성능을 비교하기 위하여 G.711 10msec프레임단위로 수신된 트래이스 결과를 비교데이터로 사용하여 검토를 한다.

I. 서론

음성데이터는 약 10%의 손실과 150msec이하의 전송 지연 정도는 사람이 불편을 느끼지 못한다. 인터넷전화는 이러한 실시간 음성데이터의 송수신을 위한 Real-Time Transfer Protocol(RTP)로 UDP를 하기 때문에 패킷의 손실이 생기게 되고 여러 네트워크를 통하여 전송이 되면서 지연과 지연의 편차를 발생하게 되어 일반전화망

II. 인터넷 전화

현재 사용되는 일반통신망을 완전히 패킷 네트워크로 교체하기에는 상당한 시간이 소요가 될 것이지만 이미 IP(Internet Protocol)는 음성,데이터, 그리고 멀티미디어 네트워크의 미래를 대표하는 프로토콜로 자리잡고 있으며 인터넷 전화 또한 일반전화를 대체하게 될 것이라는 전망이 점점 현실이 되어 가고 있다[2][3].

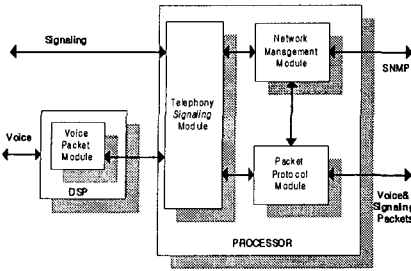


그림 3. 인터넷전화 기본 구조

인터넷전화의 기본적인 음성데이터의 처리를 위한 구조는 그림1과 같다. 일반적으로 네트워크에서는 데이터의 전송중에 지연이 발생할 수 있고, 이러한 지연은 다양한 편차를 가지고 수신단까지 도착을 하게 된다. 그림 2는 송수신되는 음성패킷의 생성과 재생관계를 통하여 지터버퍼의 필요성을 보인 것이다.

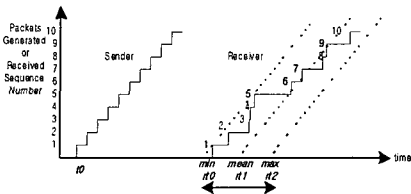


그림 4. 음성패킷의 생성과 재생

그림2에서 각 파라미터는 다음과 같다.

- t_0 : sender가 패킷을 보낸 시점
- rt_0 : receiver가 첫 번째 패킷을 받은 시점
- rt_1 : receiver에 도착한 패킷들의 지연평균
- rt_2 : receiver에 도착한 패킷의 최대 지연

그림2에서와 같이 데이터의 지연은 각각의 패킷마다 달라지게 되고 이때 패킷들은 순서까지 뒤바뀔 수가 있으며 이러한 현상을 지터(Jitter)라고 하고, 이 지터 현상에 대응하기 위한 버퍼를 지터버퍼(Jitter Buffer)라고 말한다[1][2].

III. QoS를 위한 지능적인 방식

네트워크를 통하여 이루어지는 데이터의 전송은 지연과 지연편차가 발생하므로 수신단에서는 이를 대응할 적절한 버퍼관리를 하여야 한다. 네트워크의 전송과정을 간단하게 모델링하여 보면 그림3과 같다.

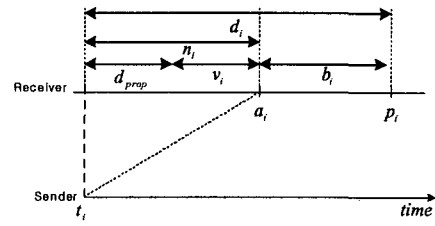


그림 5.패킷 i의 지연관계

Ramjee는 그림3과 같은 모델링을 통하여 파라미터를 결정하고 4가지의 알고리즘을 제안하였는데, 각 알고리즘은 지연평균의 계산 방법이 각기 다르다. 이들 알고리즘은 각각 평균을 계산하여 평균변화율을 계산하여 버퍼의 최대값을 고려하고, 네트워크상황을 판단하였다 [1][2]. 이는 패킷의 전송지연 시간을 계산한다고 가정하였다. 또 다른 방식으로 Jonathan 등이 [2]에서 제시한 알고리즘은 손실의 보상을 위하여 Forward Error Correction (FEC)를 도입하였다. FEC는 기본적으로 UDP가 TCP와 같이 재전송에 의한 무손실 보장이 되지 않으므로 보내는 실제 데이터의 정보를 갖는 추가 정보를 보내어 수신단에서 손실의 보상을 처리 하는데 상당히 장점을 갖는 알고리즘을 제안 하였으나 이는 추가 정보를 위한 전송포트를 하나 더 열어야 하는 문제가 있어서 실제 적용이 힘들다[3].

이에 본 논문에서는 기존의 방식들과 달리 QoS를 위하여 현재 단말기로 수신되어 오는 데이터만을 고려하여 지능적인 패킷관리 방식으로 다음의 두 가지를 고려한다.

- 1) 패킷의 도착 시점에서 이전 도착패킷과의 순서 관계
- 2) 현재 버퍼의개수와 도착시점의 관계

이 사항들을 네트워크의 상황으로 고려하는 이유는 LAN상황에서는 특별한 부하가 없는 경우 지터에 대한 대응은 단순한 버퍼로 처리가 되지만, 실제 인터넷전화 가 사용이 되는 WAN상황에서는 여러 네트워크를 거치면서 상이한 라우팅 경로차이 등으로 인한 패킷의 지연 편차와 순서 뒤바뀔 현상들이 발생하게 되는데, 이들의 검출이 네트워크 상황판단의 중요한 부분이 되기 때문이다.

제안한 방식에서 평균의 계산은 식(1)과 같고, 평균편차의 계산은 식(2)와 같다.

$$D_i = T_i - BsTime - (ArvCnt + S_B - 2S_i - 1) \times EncTime \quad ; ArvCnt \leq S_i$$

$$D_i = T_i - BsTime - (S_i - S_B) \times EncTime \quad (1) \quad ; ArvCnt > S_i$$

$$d_i = \beta \times d_{i-1} + (1 - \beta) \times D_i \quad ; D_i > d_{i-1}$$

$$d_i = \alpha \times d_{i-1} + (1 - \alpha) \times D_i \quad ; D_i \leq d_{i-1}$$

(2)

단, 식(1), (2)에서 사용된 파라미터는 다음과 같다.

- d_i : i 번째 패킷까지의 평균편차
- α, β : 평균을 구하기 위한 가중치
- T_i : i 번째 패킷의 도착시간
- D_i : i 번째 도착한 패킷의 지연시간
- $ArrCnt$: 현재까지 도착한 패킷의 개수
- $EncTime$: 압축알고리즘 인코딩 지연
- $BsTime$: 기준 패킷의 도착시간
- S_B : $BsTime$ 으로 설정된 패킷의 순서 번호
- S_i : i 번째 패킷까지의 평균편차

제안한 방식에서는 적절한 지연 평균 변화율에 따라 자동적으로 버퍼의 수를 결정하기 위하여 퍼지 시스템을 이용한다. 식(2)에서 D_i 가 버퍼링 시간 보다 짧은 경우 i 번째 패킷은 정상적인 데이터로 재생이 되므로 α 의 값은 크게 중요하지 않다. 식(1)에서 β 의 값은 고정하는 방식보다는 퍼지 시스템을 이용하여 자동적으로 조절하여 버퍼의 수를 결정하는 것이 좋다. 퍼지 시스템의 입력은 $D_i - d_{i-1}$ 과 $\Delta(D_i - d_{i-1})$ 의 변화분을 이용하였으며 이에 대한 소속 함수는 그림 4이다. 그림 5는 퍼지 시스템의 출력인 β 의 소속 함수 이고, 표1은 퍼지 rule base이다. 퍼지화는 singleton 퍼지화, 퍼지 추론은 min-max추론, 비퍼지화는 무게중심법을 이용하였다[4].

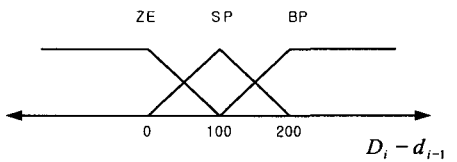


그림 4. 입력에 대한 소속 함수

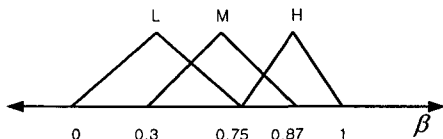


그림 5. 출력에 대한 소속 함수

표 1. rule base

$D_i - d_{i-1}$	ZE	SP	BP
$\Delta(D_i - d_{i-1})$	N	Z	P
N	H	H	M
Z	H	M	L
P	M	L	L

III. 실험 및 검토

본 논문에서 제안한 방식의 유용성을 확인하기 위하여 다양한 네트워크 상황에서의 수신 단말기 트래이스를 기존의 버퍼 관리방식과 네트워크 상황적응에 대한 응답성을 실험한다. 기존의 방식들은 목음 압축을 사용하기 때문에 재생지연의 버퍼지연시간을 같은 상황에서 비교할 수 없으므로 재생과정을 통한 손실과 재생지연은 비교하지 않고 각 알고리즘의 버퍼계산을 위한 공통부분인 지연평균의 비교를 한다. 그림6은 최대 15, 40, 70, 110, 150, 180 msec의 지연 편차가 발생하는 상황에서 수신된 패킷의 트래이스로 그림7~10에 사용된 100개씩의 샘플데이터의 일부이다. 기존의 버퍼관리과 차이를 절대적으로 하기 위하여 파라미터 값 $\alpha = 0.998002$ 을 사용하였다.

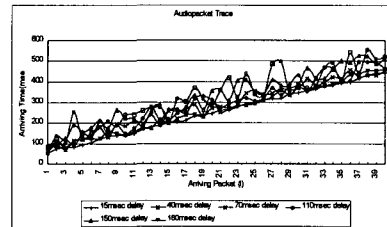


그림 6. 다양한 음성패킷트래이스

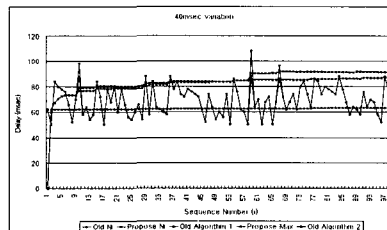


그림 7. 40msec지연변동의 네트워크

참고문헌

- [1] R. Ramjee, Jim Kurose, D. Towsley, and H. Schulzrinne, "Adaptive playout mechanisms for Packetized audio applications in wide-area networks," in the Proceedings of the Conference on Computer Communications (IEEE Inform), Toronto, Canada, June 1994, pp680-668
- [2] Jonathan Rosenberg, Lili Qiu, and Henning Schulzrinne "Integrating Packet FEC into Adaptive Voice Playout Buffer Algorithms on the Internet", in the Proceedings of the IEEE Inform 2000, Tel Avid, Israel, March 2000.
- [3] W. Ricard Stevens "TCP/IP Illustrated, Volume1" Addison Wesley
- [4] Li-Xin Wang " A course in Fuzzy system and control" Prentice-Hall , 1997

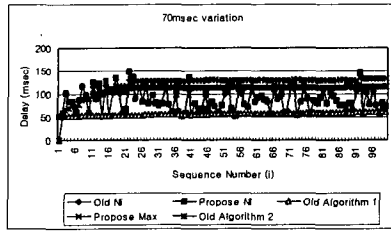


그림 8. 70msec지연변동의 네트워크

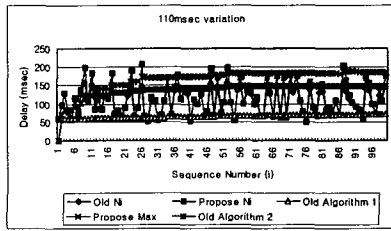


그림 9. 110msec 지연변동의 네트워크

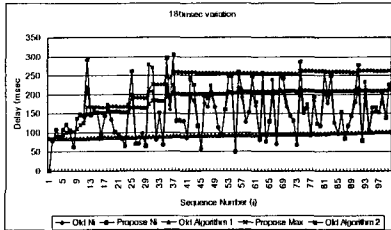


그림 10. 180msec 지연변동의 네트워크

그림 7~10에서 제안한 알고리즘이 기존의 알고리즘보다 네트워크의 변화에 보다 빠르고 정확한 네트워크 상황 안정화가 이루어 지는 것을 볼 수 있다.

V. 결 론

본 논문에서는 인터넷전화에서 보다 양질의 음성을 제공하기 위하여 단말기 입장에서 지능적인버퍼관리 방식을 제안하였다. 제안한 방식은 퍼지 시스템을 이용하여 자동적으로 버퍼의 수를 결정하여 현재 단말기에서 사용하고 있는 압축알고리즘의 종류와 수신된 데이터만을 참조하여 네트워크의 변화에 빠른 응답성을 갖는 지능적인 버퍼 관리 방식이다. 제안한 방식의 유용성을 확인하기 위하여 기존의 버퍼 관리 방식과 다양한 네트워크상황의 트레이스 결과를 비교검토 하였다. 그 결과 실질적으로 네트워크 상황에 대한 변화를 기존의 버퍼 관리 방식보다 빠른 응답성과 안정성을 보였다.