Journal of The Korea Institute of Information Security & Cryptology(JKIISC) ISSN 1598-3986(Print)
VOL.23, NO.5, October 2013 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2013.23.5.867

이벤트 방식 지터 버퍼 알고리즘의 분석*

최 승 한,[1†] 박 종 민,[2] 서 창 호[1‡]
[1]공주대학교, [2]조선이공대학교

The Analysis of Event-based Jitter Buffer Algorithm*

Seung-Han Choi,[1†] Jong-Min Park,[2] Chang-Ho Seo[1‡]
[1]Kongju National University, [2]Chosun College of Science & Technology

요 약

본 논문은 VoIP(Voice over IP) 서비스에서 사용자의 체감 음성 품질을 결정하는 중요한 요소에 해당되는 지터와 지터를 제거하기 위한 지터 버퍼 알고리즘에 대해서 설명한다. 지터 버퍼의 종류는 크게 고정형(Fixed) 지터 버퍼와 적응형(Adpative) 지터 버퍼, 두 종류로 나누어지며, 적응형 지터 버퍼는 다시 타임 방식과 이벤트 방식으로 다시 나누어지는데, 지터 버퍼 알고리즘의 분석을 통해서 성능 향상 방안을 제안한다.

ABSTRACT

In this paper, a major factor in determining voice quality that corresponds to the jitter and jitter buffer algorithm for removing jitter will be described. We analyze various jitter buffer algorithms and suggest ways to improve performance of jitter buffer algorithm.

Keywords: VoIP, Jitter Buffer, Event, De-jitter

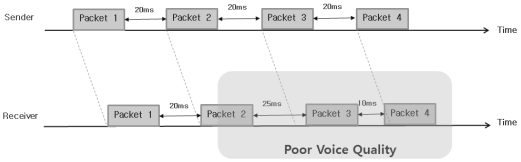
I. 서 론

최근 국내에서 LTE(Long Term Evolution) 서비스가 상용화된 이후 가장 주목받고 있는 것이 VoLTE(Voice over LTE) 이다. VoLTE는 LTE 환경에서 음성 서비스를 의미하며, 이는 음성과 데이터 통신이 모두 인터넷 프로토콜(IP)방식에서 제공될 수 있다는 의미이며 동시에 진정한 All-IP 시대의 시작을 의미한다. VoLTE를 통한 음성 서비스를 제공할 때에, 사용자의 체감 음성 품질은 중요하다[1]. VoIP(Voice over IP) 환경에서 이 음성 품질에 영향을 주는 기술 요소들은 패킷 손실(Loss), 패킷 지연 분포(지터:Jitter), 단말간 네트워크 지연(Delay), 코덱의 음성 품질, 에코(Echo) 등이 있다. 본 논문에서는 음성 품질에 영향을 주는 지터(Jitter)의 정의와 지터를 감소시키기 위한 지터 버퍼 알고리즘의 종류를 소개하며, 특히 이벤트 방식 지터 버퍼 알고리즘에 대한 분석을 통해서 지터 버퍼 알고리즘의 성능 향상 방안을 제안한다.

접수일(2013년8월6일) 게재확정일(2013년9월8일)
* This work was supported by NRF grant funded by the MEST (R01-2011-0029927)
† 주저자, shchoi@kongju.ac.kr
‡ 교신저자, chseo@kongju.ac.kr(Corresponding author)

II. 지터의 정의

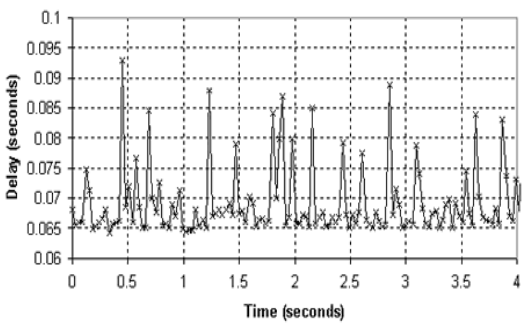
[그림 1]은 음성 패킷의 지터 발생을 도시한 그림이다. VoIP(Voice over IP) 음성 서비스 환경에서 음성 패킷은 송신측에서 20 ms 간격으로 출발하게 되며, 수신측에 20 ms 간격으로 도착해야 사용자 입장에서 음성 품질이 좋다고 느끼게 된다. 하지만, IP 망에서 여러 가지 이유로 지연이 발생해서 음성 패킷이 20 ms 간격으로 수신측에 도달하지 못하면, 사용자는 음성 품질이 나쁘다고 판단한다. 이처럼 음성 패



〔그림 1〕 음성 패킷의 지터 발생

킷의 지연 차이를 지터(Jitter) 라고 한다. 좀 더 정확히 표현하면, 음성 패킷 도착 시간 사이의 통계적 변화량(Statistical variance)을 지터라고 한다. 지터가 발생하게 되는 원인은 네트워크 장비에서의 큐잉(Queuing)/버퍼링(Bufferring), 패킷 재라우팅, 네트워크 멀티플렉싱(Multiplexing), 타이밍 드리프트(Timing Drift) 등 여러 가지가 존재한다.

〔그림 2〕는 랜(LAN) 환경의 패킷 정체(Congestion)에 의해서 발생하는 패킷의 지연을 도시한 그림이다〔2〕〔7〕. Congestion 의 의해서 패킷의 지연 차이가 발생하는데, 이것은 수신측에서 지터를 의미한다.



〔그림 2〕 랜(LAN)망의 Congestion에 의한 지연 변화

①

$$D(i,j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i)$$

②

$$J(i) = J(i-1) + (|D(i-1,i)| - J(i-1))/16$$

〔그림 3〕 지터 계산 공식

〔그림 3〕는 지터를 계산하는 공식을 도시한 그림이다〔3〕. ①식은 두 개 패킷(i번째/j번째 패킷:j>i)의 송신 시간과 수신 시간을 차이를 계산한 식이고, ②식은 계산된 ①식을 이용한 i번째 패킷을 수신한 이후의 지터를 계산한 식이다. 이전 (i-1)번째 패킷을 수신 했을때의 지터를 16으로 나눈 것은 노이즈(잡음)를 줄이기 위한 것이다.

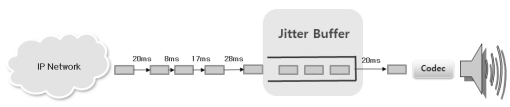
〔표 1〕 공식에 의한 지터 계산 예제

I	S_i	R_i	D(i-1, i)	J(i)
1	0	10	0	0
2	20	30	0	0
3	40	49	-1	0.0625
4	60	74	5	0.3711
5	80	90	-4	0.5979
6	100	111	1	0.6230
7	120	139	8	1.0841
8	140	150	-9	1.5788
9	160	170	0	1.4802
10	180	191	1	1.4501
11	200	210	-1	1.4220
12	220	229	-1	1.3956
13	240	250	1	1.3709
14	260	271	1	1.3477

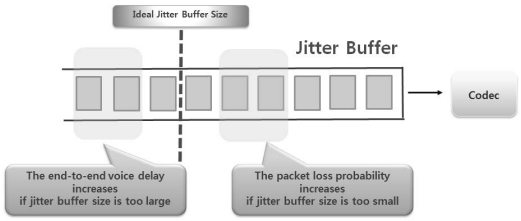
〔표 1〕은 지터 계산 공식에 의해서 계산한 예제 테이블이다. 3번째 패킷이 수신되었을 때부터 지터 가 발생한 상태를 확인 할 수 있다. 지터는 사용자의 음성 품질을 저하 시키는 원인이 되며, 지터를 최소한으로 줄여서 음성 품질을 향상 시키기 위한 방법으로 지터 버퍼를 두고 있으며, 지터 버퍼를 통해서 지터를 제어하기 위한 알고리즘을 디지터(De-jitter), 또는 지터 버퍼 알고리즘이라고 한다. 다음 장에서 지터 버퍼 알고리즘에 대해서 자세히 소개 하도록 한다.

III. 지터 버퍼의 종류

〔그림 4〕는 지터 제거를 위한 지터 버퍼의 역할을 도시한 그림이다. 송신측에서 음성 패킷을 20ms 단위로 송신했지만, 인터넷망을 통과하는 동안 패킷과 패킷의 간격이 20ms 단위로 유지가 되지 않고 빠르거나, 느리게 차이가 발생하게 된다. 지터를 제거하기 위한 방법으로 수신측에 지터 버퍼를 사용하여 20ms 단위로 도착하지 못한 음성 패킷을 버퍼에 저장한 후에, 코덱 디코딩 모듈로 20ms 단위로 균일한 시간 차이로 보내게 된다. 즉, 버퍼를 이용해서 지터를 제거하여 사용자의 귀에 들리도록 한다.



〔그림 4〕 지터 제거를 위한 지터 버퍼

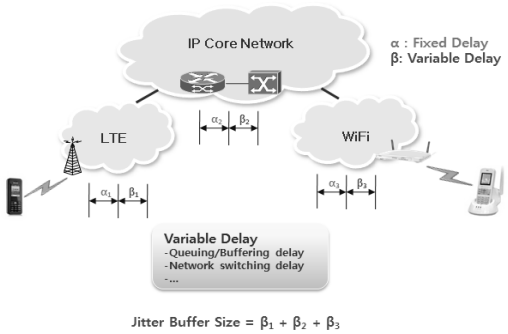


〔그림 5〕 지터 버퍼의 구조

〔그림 5〕는 지터 버퍼의 구조를 도시한 그림이다. 지터를 제거하기 위해서 지터 버퍼의 크기를 너무 크게 정하거나, 또는 너무 작게 정하면, 문제가 발생한다. 지터 버퍼의 크기가 너무 크면, 패킷이 버퍼에 머무르는 시간이 늘어나기 때문에 단대단(End-to-End) 지연이 그 만큼 증가한다. 반대로, 지터 버퍼 크기가 너무 작으면, 버퍼가 모두 차게 되어 패킷이 손실(Loss) 되는 문제가 발생한다. 즉, 네트워크 상태, 단말기 상태 등에 맞는 적절한 지터 버퍼 사이즈가 결정되어야 하며, 이를 위해서 지터 버퍼 알고리즘이 필요하다.

〔그림 6〕은 네트워크상의 지연과 지터 버퍼의 크기의 관계를 도시한 그림이다. 결국, 고정된 지연은 지터를 제거하기 위한 지터 버퍼의 크기와는 관계가 없고, 가장 이상적인 지터 버퍼의 크기는 가변 지연의 총합이며, 가장 우수한 지터 버퍼 알고리즘은 네트워크상의 가변 지연을 최대한 정확히 찾아내는 알고리즘이라고 볼 수 있다.

지터 버퍼의 종류는 크게 고정형(Fixed) 지터 버퍼와 적응형(Adpative) 지터 버퍼, 두 종류로 나누어 진다. 고정형 지터 버퍼는 구현이 간단하고, 단말



〔그림 6〕 가변 지연과 지터 버퍼 크기의 관계

시스템에 오버헤드가 적은 장점이 있다. 하지만, 네트워크의 가변 지연을 반영하지 못하기 때문에 지터 버퍼의 크기를 너무 작게 설절하면, 지터 버퍼가 모두 차게 되어 패킷이 손실되고, 지터 버퍼의 크기를 너무 크게 설정하면, 지연이 커지는 단점이 있다.

적응형 지터 버퍼는 다시 타임 방식과 이벤트 방식, 두 가지 종류의 지터 버퍼 알고리즘을 수행하는 지터 버퍼로 나누어 진다. 먼저 타임 방식의 적응형 지터 버퍼는 RTP〔4〕 패킷의 타임스탬프와 같은 정보를 이용해서 네트워크의 가변 지연 상태를 유추하여 지터 버퍼 크기를 계산하는 알고리즘을 수행하여 지터 버퍼를 동작 시킨다. 이것은 네트워크의 상태를 유추하여 지터 버퍼 크기의 계산하기 때문에, 고정형 지터 버퍼에 비해서 보다 적절한 지터 버퍼 크기를 찾아내는 장점이 있지만, 지속적으로 네트워크 상태를 이용해서 지터 버퍼 크기를 계산하는 방법이 복잡하여 단말 시스템에 오버헤드가 되고, 단말 시스템 마다 클럭이 다르기 때문에 가변 지연을 계산하는데 오차가 발생하는 단점이 있다.

① $D_{i,\min} = \underset{j}{Min}\{D_{j,\neq t}\}_{j=1,\ldots,i-1}$

② $B_{i-1} = D_{i-1} - D_{i,\min}$ for $i > 1$

③ $B_i^{'} = B_{i-1}(1+f)$ $B_i = B_i^{'} + D_{i,\min} - D_{i+1,\min}$

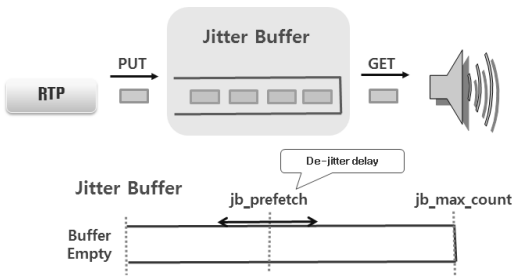
〔그림 7〕 타임 방식 적응형 지터 버퍼 알고리즘의 지터 버퍼 크기 계산 수식

〔그림 7〕은 타임 방식 적응형 지터 버퍼 알고리즘에서 사용되는 지터 버퍼 크기 계산 수식이다〔5〕. ①식의 $D_{i,\min}$는 I번째 까지 수신된 패킷들의 네트워크 지연 중에 가장 작은 네트워크 지연을 나타내는데 이것은 고정 지연을 의미한다. ②식에서 i-1번째 패킷의 네트워크 지연에서 고정 지연을 뺀 값을 지터 버퍼의 크기로 B_{i-1}으로 정했는데, 결국 이것은 네트워크 가변 지연을 의미한다. ③식은 B_{i-1}값에 인자 f를 추가하여 초기 B_i값인 $B_i^{'}$를 도출하고, $B_i^{'}$에 가변 지연을 더해서 최종적인 지터 버퍼 크기인 B_i를 도출한다.

다음, 이벤트 방식의 적응형 지터 버퍼 알고리즘은 수신 패킷의 코덱 프레임이 지터 버퍼에 입력되고, 출력되는 개수를 이용해서 지터 버퍼의 크기를 결정하는 알고리즘으로서, 다음 장에서 자세히 소개하도록 한다.

IV. 이벤트 방식 지터 버퍼 알고리즘의 분석

〔그림 8〕은 이벤트 방식의 적응형 지터 버퍼의 구조를 도시한 그림이다〔6〕. RTP 패킷에 포함된 코덱 프레임이 지터 버퍼에 들어가는 동작을 PUT으로 정하고, 지터 버퍼에 저장 되었던 코덱 프레임이 다시 지터 버퍼에서 출력되는 동작을 GET 동작을 정한다. GET과 PUT 동작을 각각 카운트 해서 GET과 PUT의 개수가 같으면(+1/-1 차이는 허용), 지터가 발생하지 않은 것으로 간주하여, 기존의 지터 버퍼 크기(jb_prefetch)를 그대로 유지한다. 개수에 차이가 발생하면 지터가 발생한 것으로 간주하여 지터 버퍼 크기 계산 로직을 이용해서 다시 지터 버퍼 크기를 결정하게 된다.



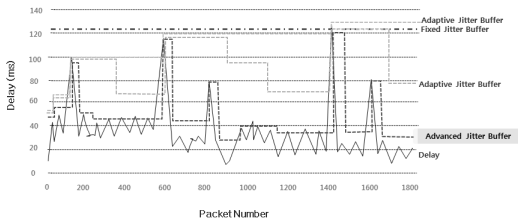
〔그림 8〕 이벤트 방식 적응형 지터 버퍼의 구조

이벤트 방식의 지터 버퍼 구조에서 지터 버퍼 크기를 계산하는 알고리즘은 다음과 같다. PUT과 GET의 차이를 계산해서 기존 지터 버퍼 크기보다 작은 상태가 일정한 횟수만큼 반복되면, 현재 지터 버퍼의 크기를 조정하게 된다. 기존 지터 버퍼 크기에서 PUT과 GET의 차이가 가장 컸던 값의 차이를 3으로 나누어서 최종 차이를 계산하고, 다시 기존 지터 버퍼 크기에서 최종 차이를 빼서 새로운 지터 버퍼 크기를 계산해서 적용한다. 한편, PUT과 GET의 차이를 계산해서 기존 지터 버퍼 크기보다 큰 상태가 되면, 역시 현재 지터 버퍼의 크기를 조정하게 된다. PUT과 GET의 차이가 가장 컸던 값의 차이와 물리적인 지터 버퍼 크기에 가중치를 적용한 값에서 가장 작은 값을 새로운 지터 버퍼 크기를 계산한다.

이벤트 방식의 지터 버퍼 구조는 타임 방식의 지터 버퍼 구조에 비해서 단순한 알고리즘으로 동작하기 때문에, 시스템의 오버헤드가 적어서 스마트폰과 같은 모바일 단말의 VoIP 미디어 엔진에 적용하기에 적합한 구조이다.

V. 결 론

지터 버퍼의 종류는 크게 고정형(Fixed) 지터 버퍼와 적응형(Adpative) 지터 버퍼, 두 종류로 나누어 진다. 적응형 지터 버퍼는 다시 타임 방식과 이벤트 방식, 두 가지 종류의 지터 버퍼 알고리즘을 수행하는 지터 버퍼로 나누어 진다. 이벤트 방식의 지터 버퍼 구조는 타임 방식의 지터 버퍼 구조에 비해서 단순한 알고리즘으로 동작하기 때문에, 시스템의 오버헤드가 적은 장점을 가지고 있다.



〔그림 9〕 지연과 지터 버퍼 알고리즘의 관계

〔그림 9〕는 지연과 지터 버퍼 알고리즘의 상관 관계를 도시한 그림이다. 고정형 지터 버퍼는 패킷 지연 차이에 상관없이 지터 버퍼 크기라 정해져 있기 때문에 그림처럼 지연 버퍼에 의한 지연이 항상 존재한다. 기존의 적응형 지터 버퍼는 패킷의 지연 정보를 이용해서 지터 버퍼의 크기를 가변적으로 조정하기 때문에, 고정형 지터 버퍼보다 지터 퍼버 지연이 줄어들게 되어 음성 품질에 좋은 영향을 주게 된다. 하지만, 〔그림 9〕에서처럼 가변적으로 변하는 패킷의 지연 차이를 적응형 지터 버퍼 알고리즘도 비교적 정확하게 지터 버퍼 크기를 결정하지 못하는 문제가 발생하여, 지터 버퍼 지연이나 지터 버퍼에서의 패킷 손실이 발생한다. 즉, 기존의 적응형 지터 버퍼 알고리즘을 개선한 보다 향상된 지터 버퍼 알고리즘(Advanced Jitter Buffer)의 연구가 필요하다.

참고문헌

〔1〕 최택진 LG유플러스 기술부장, “VoLTE 시대 눈앞…무엇이 달라지나,” KAIT 웹매거진, 11호, pp. 28-31, 2012년 5월

〔2〕 “Indepth Article: Jitter ,” http://www.voiptroubleshooter.com/indepth/jittersources.html.

〔3〕 Vladimir Tincar, "VoIP Basics: About Jitter", http://toncar.cz/Tutorials/VoIP/VoIP_Basics_Jitter.html.
〔4〕 H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," IETF RFC 3550, Jul. 2003.
〔5〕 Gagan L. Choudhury, Robert G. Cole, "Design and analysis of optimal adaptive de-jitter buffers," Computer Communications, vol. 27, Issue 6, pp. 529-537, Apr. 2004.
〔6〕 "Adaptive Jitter Buffer," http://www.pjsip.org.
〔7〕 신상욱, 이경현, "안전한 데이터 통신에서의 지연 분석," 한국정보호호학회 논문지, 7(4), pp. 23-26, 1997년 12월

〈저자소개〉

최 승 한 (Seung-han Choi) 일반회원
1998년: 충남대학교 컴퓨터공학과 학사 졸업
2000년: 충남대학교 컴퓨터공학과 석사 졸업
2000년~현재: 한국전자통신연구원 선임연구원 재직
2011년~현재: 공주대학교 융합과학과 박사 과정 재학중
〈관심분야〉 통신공학, 임베디드 소프트웨어, 멀티미디어 프로토콜, 분산 시스템

박 종 민(Park-Jong Min) 정회원
1986년~1997년: 기아정보시스템(주)
1988년: 조선대학교 전자계산 전공(공학석사)
2005년: 조선대학교 컴퓨터공학 전공(공학박사)
2008년~현재: 조선이공대학교 컴퓨터보안과 교수
2010년~현재: 한국멀티미디어학회 이사
〈관심분야〉 정보보호 및 보안, 네트워크 보안

서 창 호(Seo-Chang Ho) 종신회원
1990년: 고려대학교 수학과 졸업(학사)
1992년: 고려대학교 수학과(이학석사)
1996년: 고려대학교 수학과(이학박사)
1996년~1996년: 국방과학연구소 선임연구원
1996년~2000년: 한국전자통신연구원 선임연구원, 팀장
2000년~현재: 공주대학교 융합과학과 교수
〈관심분야〉 암호 알고리즘, PKI, 무선 인터넷 보안