Enhanced Playout Buffer Algorithm for VoIP based Social-Network Applications

Neha Tiwari
The IIS University Jaipur
E-mail: sneha.sh02@gmail.com

O.P.Rishi
Central University of Rajasthan
E-mail: omprakashrishi@yahoo.com

Abstract:

Playout buffer of Voice over Internet Protocol (VoIP) applications play an important role in packet rescheduling and playing them back in sequence. VoIP implementation suffers from impairments like Delay, Jitter, Spiky delay, Packet loss, Acoustic echo and Packet sequence. Many Social Network applications are offering VoIP facility with them but are unable to cope up with the impairments associated with it and delivering better Quality of Service (QoS) to the users. There is a need to analyze the types of algorithms and their relevance for social network applications. This paper offers an analysis of the types of playout buffer algorithms, models available for evaluating perceived speech quality and finding out the best among them. It also highlights the need of fine-tuning of VoIP based social network applications.

Keywords: ***social networks, VoIP, VoIP impairments, playout buffer, playout buffer algorithms, voice quality assessment models, delay, jitter, spiky delay, packet loss*.**

1. Introduction

Social networks have given a new meaning to the world of Internet. They are offering multiple tools and applications for people to make their conversation more interactive, and VoIP is one of those. VoIP helps in having voice communication over packet switched Internet Protocol (IP) network. Since Internet was primarily developed for the transfer of data and not for voice, there are a few factors that sternly affect the quality of voice. Unlike circuit switched Public Switched Telephone Network (PSTN), in packet switched network packets reach the destination at different time and in different order, therefore there is a need of some mechanism to hold the packets, reorder them and play them as a single stream. This is achieved by a buffer called as “playout buffer” or “jitter buffer” [1]. The most decisive role of this buffer is to maintain a trade-off for jitter, packet loss, voice quality and user interactivity. A large sized playout buffer would definitely offer a good quality of voice but will compromise with user interactivity and vice versa, therefore a good playout buffer must keep a balance between voice quality and user interactivity. There are certain algorithms that help Playout buffer in achieving its objectives.

2. Playout Buffer Adjustment Problem :

Buffer size adjustment is an optimization problem that has had gathered much attention of researchers over the past few years. Many researches have been going on for the improvement of buffer size adjustment. Theoretically different types of playout buffer algorithms have been proposed that help in fine-tuning the playout buffer but the practical implementation of these algorithms in different situations on the basis of different functional parameters of performance and quality of voice needs to be analyzed, to identify limitations and advantages of these algorithms.

Refining playout buffer functionality is inseparable from the user satisfaction. Models are available for evaluating voice quality but the testing technique(s) and approach(es) for different model is different and moreover every model works on different parameter(s) and different concept(s). To evaluate the perceived speech quality we need to have a standardized model to evaluate functionality of playout buffer.

3. Playout buffer algorithms:

Playout buffer performs two tasks reordering the packets arriving at receivers end and holding the packets till their scheduled playout time. Following are the types of terms associated with the i^{th} packet during transmission from sender to receiver [2]. Here, t_i is the sending time, a_i is the arrival time, n_i is the total network delay, b_i is buffer delay, p_i is the playout time which is the time at which the packets are actually being played and d_i is the total end-to-end delay also called as "playout delay".

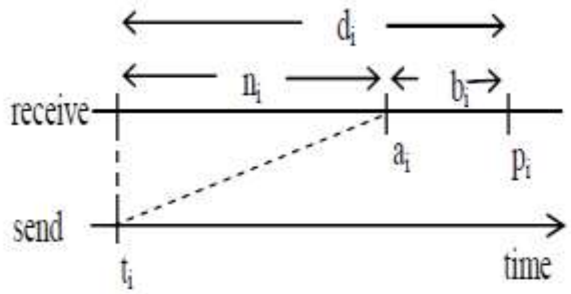

Figure1. [2] Timing associated with packet "i"

The playout buffer algorithms can be categorized as "fixed" and "adaptive". Fixed playout buffer algorithms are for fixed size buffer that does not adjust its size, while adaptive buffer algorithms help the playout buffer in adaptively adjust its size according to varying network conditions [3].

Adaptive buffer helps in maintaining jitter, packet loss, voice quality and user interactivity. On the basis of different roles played by jitter buffer algorithms we can classify it in following types [4]:

i. Reactive algorithms – perform continuous estimation of network delays and jitter to calculate playout deadlines. [3] [5] [6]

ii. Histogram-based algorithms – maintain a histogram of packet delays and choose the optimal playout delay from that histogram. [7] [8]

iii. Algorithms that monitor packet loss ratio or buffer occupancy and adjust the playout delay accordingly. [9]

iv. Algorithms that aim in maximizing user satisfaction. [2] [10] [11]

There are multiple statistical methods that help in finding out better algorithm on the basis of delay and packet loss percentage but they do not provide any mechanism to test their validity for delivering better voice quality.

4. Playout Buffer adjustment & Voice quality measurement techniques:

Adjusting playout buffer size will not be of much use without taking user satisfaction in consideration. User satisfaction can be counted in terms to things quality and interactivity. Measurement of the quality of voice can be done either through "subjective tests" or "objective tests" [12] [13].

Subjective tests are also called as listening-only tests which are conducted in laboratories. In this traditional system the perceived voice quality is defined according to a 5-grade scale known as "mean listening-quality opinion score", commonly known as "Mean Opinion Score" (MOS) [12]. Though this method is the most authentic method but its time consuming, costly, difficult to repeat and rarely gives identical results. Moreover, it does not consider delay impairments, therefore cannot be used to assess user interactivity.

Objective tests analyze the distortion of voice signals that travel through a VoIP network. Audible error is estimated by subtracting an examined and a reference voice signal and then mapping the result to the MOS scale. This testing technique is called as "Perceptual Speech Quality Measure" (PSQM) which due to certain limitations in specific areas was replaced by "Perceptual Evaluation of Speech Quality" (PESQ) [14]. This method is designed only for one-way "listening-only" quality measurement and entails a reference speech signal. Since this method does not include delay impairments, it's not recommended to assess the end-to-end conversational call quality.

E-model [13] is a tool that estimates end-to-end voice quality by taking VoIP parameters and impairments into account. This method combines individual impairments due to both signal's properties and network characteristics into a single R-rating. High R values in a range of $90<R<100$ are considered as excellent quality and low value as low quality.

E-model does not take into account the dynamics of a transmission due to adaptive Playout buffering and relies only on static transmission impairments like average delay, average packet loss etc. On the contrary, PESQ considers playout adaptation but does not consider absolute delay into its rating.

Therefore, a combination of both the methods can accurately and efficiently evaluate the conversational speech quality. However such combination of methods does not work in real time and require a reference speech signal.

5. Comparing basic playout buffer algorithms:

Ramjee et al. (1994) [3] evaluated the effect of four basic adaptive algorithms using experimentally obtained delay measurements of audio traces between several different Internet sites. They pointed the host level issues of how to adaptively respond to the variable delays incurred as packets traverse the network.
The idea behind all these algorithms is based on "absolute timing methods" as defined by Montgomery. To calculate the playout point for packet i we have to cases:

i. If i is the first packet of talkspurt then its playout time will be calculated as
$p_i = t_i + \hat{d}_i + 4 * \hat{v}_i$
Here $\hat{d}_i$ and $\hat{v}_i$ are estimates of the mean and variation in the end-to-end delay during the talk-spurt.

ii. The playout time for subsequent packets in a talk-spurt is computed as
$p_j = p_i + t_j - t_i$

Though $\hat{d}_i$ and $\hat{v}_i$ are computed for every received packet they are only used to determine the playout point for the first packet in any talkspurt.
The basic four algorithms differ only in the manner in which $\hat{d}_i$ is calculated whereas $\hat{v}_i$ is computed in same manner for all the algorithms as proposed in Algorithm 1.
According to Algorithm 1 ("exp-avg") the delay estimate of the i_{th} packet is calculated using RFC793 algorithm and a measure of the variation in delays is calculated as suggested by Van Jacobson [15]. The mean delay is estimated through an exponentially weighted average.
$\hat{d}_i = \alpha * \hat{d}_{i-1} + (1 - \alpha) * n_i$
Whereas variation is computed as
$\hat{v}_i = \alpha \hat{v}_{i-1} + (1 - \alpha)|\hat{d}_i - n_1|$
Algorithm 2 ("fast-exp") is a small modification to first algorithm as suggested by Mills [16]. It helps in adapting more quickly to the short burst of packets incurring long delays. It uses smaller weighting factor as delays increase

$$\hat{d}_i = \begin{cases} \beta\hat{d}_{i-1} + (1-\beta)n_i & n_i > \hat{d}_{i-1} \\ \alpha\hat{d}_{i-1} + (1-\alpha)n_i & n_i \le \hat{d}_{i-1} \end{cases}$$

Algorithm 3 ("min-delay") is the delay adaptation algorithm and is more aggressive in minimizing delays. It uses minimum delays of all packets received in the current talkspurt. It requires less computation but doesn't perform well in cases where jitter is high. If S_i is the set of delays

$$\hat{d}_i = \min_{j \in S_i} \{n_j\}$$

Algorithm 4 ("spk-delay") is based on fast adaptation to spikes (sudden, large increase in end-to-end network delay) which first three algorithms lack. During a spike, the delay estimate tracks the delays closely, but after the spike is over this algorithm is same as algorithm 1.

Ramjee et al. [3] compared all these algorithms from the perspective of number of packets dropped and found out that Algorithm 4 outperforms algorithms 1 to 3 for both a given average playout delay and a given maximum buffer size. Their experiment showed that the "fast-exp" has the lowest loss rate and it adapts more quickly to increase in delay whereas "min-delay" has the lower delay and higher loss. The other two algorithms are between "fast-exp" and "min-delay". They also say that the algorithm that proved to be good for one domain need not necessarily be good for another domain.

Many other algorithms are also available but they are variations of these basic algorithms. Moreover, the experimental work done by taking all these basic and derived algorithms used voice traces. Real-time communication has not been used to conduct the analysis and evaluation of the performance of the algorithms.

6. Playout buffer adjustment

Most of the playout buffer dimensioning algorithms adjusts the buffer size on the basis of linear combination of network delay and jitter. Ramjee et al. [3] proposed the idea to adjust the buffer size on the basis of EWMA (Exponential Weighted Moving Average) of network delays and delay jitters but the weights of the variables are fixed and empirically chosen.

Narbutt and Murphy [6] provided a breakthrough with their work by adaptively adjusting the EWMA weights according to the magnitude of delay jitter. They proposed to set the weight smaller when the delay is higher and vice versa. They proved through their simulation work that their adaptive approach considerably improves the tradeoff between the buffer delay and packet loss.

With further elaboration to the works mentioned above, Liang et al. [17] and Sreenan et al. [18] adjusted the buffer size within a talk burst. The idea behind such improvements in playout algorithms is to help the playout buffer adapt to varying network conditions more rapidly so as to achieve a better conversational quality.

The refinement work done so far regarding the adjustment of playout buffer reveals that it is based primarily on a combination of factors mainly network delay and their standard deviations ("delay jitters"). The load of packets on VoIP based social network application is unpredictable. The buffer may suffer from either underflow or overflow. Though many alterations have been proposed to the playout buffer algorithms so as to reduce the packet loss but that has been done in context to delays and delay jitters. Since as the number of VoIP connections ("nodes" in social network terms) increase the chances of buffer overflow also increase. The overflow results in packet loss which further affects the quality of conversation voice. Therefore there is a requirement to refine the playout buffer algorithm according to the need of the VoIP based social network applications so as to trim down the packet loss and have a better voice quality.

7. Conclusion

Researches have provided many adaptive playout buffer algorithms and variation of the basic algorithms but the real implementation evidence of these is still unavailable. Since social networks suffer from the problem of fluctuating load, VoIP based social network applications have reinforce the thought of evaluating the real life VoIP applications playout buffer algorithm and their co relation with voice quality.

There is a need to find a gap between what researches say about buffer algorithms and how far the applications actually implement them. Fine tuning the buffer algorithm is required so as to achieve better voice quality and user interactivity. Furthermore, there is also an urge to come up with a model that can assess the quality of voice from a social network requirement perspective.

References:

[1] Wu, Chen-Chi, Kuan-Ta chen, Yu-Chun Chang, Chin-Laung Lei (2009), "An Empirical Evaluation Of VOIP Playout Buffer Dimensioning In Skype, Google Talk, And MSN Messenger", *Proceedings of the 18th International workshop on Network and operating systems support for digital audio and video*, Taiwan Roc Proceeding NOSSDAV '09

[2] Sun L., Emmanuel C. Ifeachor, "Prediction of Perceived Conversational Speech Quality and Effects of Playout Buffer Algorithms"

[3] Ramjee, R., J. Kurose, D. Towsley, and H. Schulzrinne (1994), "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks," 13th *Proceedings of IEEE INFOCOM, Networking for Global communications*, 12th – 16th June 1994, Toronto, Canada, vol. 2, pp. 680 – 688

[4] Narbutt M., Mark Davis (2005), "Assessing the quality of VoIP transmission affected by playout buffer scheme",

[5] Bolot, J. C. and A. Vega-Garcia, "Control mechanisms for packet audio in the Internet", Proceedings of the IEEE Infocom'96, April '96, http://citeseer.ist.psu.edu/bolot96control.html

[6] Narbutt, M., L. Murphy, "VoIP Playout Buffer Adjustment using Adaptive Estimation of Network Delays," Proceedings of the 18th International Teletraffic Congress – ITC-18, pp.1171-1180, September '03

[7] Shivakumar. N., C. J. Sreenan, B. Narendran and P. Agarwal, "The Concord algorithm for synchronization of networked multimedia streams", Proceedings of the IEEE International Conference on Multimedia Computing and Systems, May '95

[8] Moon, S. B., J. Kurose and D. Towsley, "Packet audio playout delay adjustment:

Performance bounds and algorithms", ACM/Springer Multimedia Systems, Vol. 6, January '98

[9] Stone, D. L., K. Jeffay, "An empirical study of delay jitter management policies", ACM/Springer Multimedia Systems Journal, Vol. 2, No. 6, January '95

[10] Markopoulou, A. P., "Assessing the quality of Multimedia Communications over Internet Backbone Networks", Stanford University, 2003

[11] Fujimoto, k., Shingo Ata and Masayuki Murata, "Adaptive playout buffer algorithm for enhancing perceived quality of streaming applications", Proceedings of the IEEE GLOBECOM 2002 Conference, Vol. 21, No. 1, pp. 2463-2469, November '02

[12] ITU-T P.800, Methods for subjective determination of transmission quality

[13] ITU-T Recommendation G.107 (05/2000), The E-model, a computational model for use in transmission planning

[14] ITU-T Recommendation P.862, Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs

[15] Jacobson, V., "Congestion avoidance and control," *Proc. 1988 ACM SIGCOMM Conf.*, (Aug.1988, Stanford), pp. 314-329

[16] Mills, D., "Internet Delay experiments," *ARPANET Working Group Requests for Comment*, (December 1983), RFC 889

[17] Liang, Y. J., N. Farber, and B. Girod, "Adaptive playout scheduling and loss concealment for voice communication over ip networks", IEEE Transactions on Multimedia, pp. 532-543, 2003

[18] Sreenan, C. J., J.C. Chen, P. Agrawal and B. Narendran, "Delay reduction techniques for playout buffering", IEEE Transactions on Multimedia, pp. 88-100, 2000.