Classical Playout Buffer Algorithm Revisited

Carlos A Rodríguez López[1], Vitalio Alfonso Reguera[1], Evelio M. G. Fernández[2], Walter Godoy Jr[3]

[1]Telecommunications Department – Central University of Las Villas (UCLV)
Santa Clara, Cuba

[2]Electrical Engineering Department – Federal University of Parana
Curitiba, Brazil

[3] Electrical Engineering Department
Federal University of Technology-Parana – Curitiba – Brazil

{crodrigz,vitalio}@uclv.edu.cu, evelio@ufpr.br, godoy@utfpr.edu.br

Abstract. *Playout buffer absorbs packet jitter in order to compensate network delay variations. In the last decades a great number of playout buffer algorithms have been proposed for voice over IP applications. However, only a few have received the attention of the research community. In this paper we study four variations of a classical algorithm under actual network scenarios according to recent network model proposals. The MOS scale is used to assess the playout buffer performance in networks with different impairment severities. Based on the obtained results we propose a new algorithm that outperforms its predecessors. The proposed mechanism is simple and easy to implement in real applications.*

Keywords. *Playout buffer algorithm, Jitter, Quality of Service (QoS), Voice over IP (VoIP)*

Resumo. *O buffer de playout é utilizado para compensar a variação do atraso em redes de pacotes. Nas últimas décadas, um grande número de algoritmos de buffer de playout tem sido proposto para aplicações de voz sobre IP, porém apenas algumas delas têm recebido atenção pela comunidade científica. Neste artigo estudamos quatro variações de um algoritmo clássico considerando cenários reais de rede de acordo com propostas recentes de modelos de redes. A escala MOS é utilizada para avaliar o desempenho do buffer de playout em redes com diferentes níveis de imperfeições. A partir dos resultados obtidos, um novo algoritmo é proposto que supera seus predecessores. O mecanismo proposto é simples e de fácil implementação em aplicações reais.*

1. Introduction

Years have gone of efforts to obtain Internet voice communication that offers acceptable quality to satisfy the user expectations. The traditional way of supporting voice service using circuit switched is characterized by the temporal transparency of the connections. Internet was not design thinking on isochronous services; packets suffer variable delays during the transmission.

To solve this problem in VoIP connections a playout buffer is used. It serves as a temporal store for audio packets, absorbing delay variations and allowing the reproduction of the audio payload free of jitter. This operation introduces an additional delay. To establish the buffer size is an important task that has a tremendous impact in the packet loss rate and in the interactivity of the conversation.

A considerable number of algorithms exist for adaptive playout buffer, a classification can be found in [Atzori and Lobina 2006] and [Narbutt, et al 2005]. The existing solutions have been evaluated using real audio packet traces or by means of discrete event simulations. Recently it was suggested in [ITU-T 2007] a model that statistically evaluates multimedia transmission performance over IP networks. In this paper a comparative study was carried out using packet traces generated according to [ITU-T 2007].The performance of the algorithms is evaluated using de MOS scale computed through the E-model [ITU-T 2003]. Finally we propose a new playout buffer algorithm based on a classical mechanism described in [Ramjee et al 1994]. The obtained results evidences that our algorithm outperforms its predecessors with minimal computational cost.

The remainder of this paper is structured as follows. Section 2 provides useful background for our work, including a brief description of four classical playout buffer algorithms; as well as the models used to generate audio packet traces and assesses the offered quality of service for voice communications. In Section 3 we reexamine the performance of typical playout buffers discovering an important regularity that has not been previously reported to the best of our knowledge. Based on this finding, in Section 4 we propose a new algorithm that achieves the highest score. Section 5 concludes the paper.

2. Background

2.1 Playout Buffer Algorithms

During the last decades several algorithms have been introduced for the playout buffer. One of the top referenced algorithms by the research community is the one presented by Ramjee et al (1994). This algorithm adjusts the parameters of the playouts between audio bursts. Its basic functioning can be described as follow: for every incoming packet the mean network delay ($\hat{d}_i$) and mean delay variation ($\hat{v}_i$) are estimated, if the received packet is the first one in a talkspurt then the absolute delay p_o is calculated as:

$$p_o = \hat{d}_i + \Omega \cdot \hat{v}_i \tag{1}$$

where Ω is set to 4 in the original implementation. For the *i*th packet in a talkspurt the reproduction time is set to:

$$p_i = t_i + p_o \tag{2}$$

where t_i is the time stamp set by the transmitter. Four different algorithms have been proposed to estimate the mean network delay and are summarized in Figure 1.

Algorithm 1 calculated the mean network delay using an approach suggested by [Jacobson V. (1988)] for estimating the round-trip-time in TCP connections. This

n_i : network delay for packet i
$\alpha = 0.998002$
$\beta = 0.75$

Algorithm 1

$\hat{d}_i = \alpha\hat{d}_i + (1\text{-}\alpha)n_i$

Algorithm 2

if $n_i > \hat{d}_i$
  $\hat{d}_i = \beta\hat{d}_i + (1\text{-}\beta)n_i$
else
  $\hat{d}_i = \alpha\hat{d}_i + (1\text{-}\alpha)n_i$
end

Algorithm 3

For the first packet of a talkspurt

$\hat{d}_i = n_i$

For every received packet

if $n_i < \hat{d}_i$
  $\hat{d}_i = n_i$
end

n_i : network delay for packet i
$\alpha = 0.785$

Algorithm 4

if *normal_mode*
  if $|(n_i - n_{i-1})| > 2\hat{v}_i + 800$
    var = 0
    normal_mode = false
  end
  $\hat{d}_i = \alpha\hat{d}_i + (1\text{-}\alpha)n_i$
else
  $var = var/2 + |(2n_i - n_{i-1} - n_{i-2})|/8$
  if $var <= 63$
    normal_mode = true;
  else
    $\hat{d}_i = \hat{d}_i + n_i - n_{i-1}$
  end
end

Figure 1. Pseudocode for Algorithm 1, 2, 3 and 4.

algorithm uses an exponential weighted moving average whit a weighting factor α. The second algorithm (Algorithm 2) introduces a small modification to the first algorithm, changing the weighting factor α by β (a lower one) when the actual network delay exceeds its estimate. The idea is to react more quickly when the network delay increase. Algorithm 3 stands on a different approach, setting the estimated mean network delay to the minor network delay observed during a talkspurt. Finally, Algorithm 4 is designed to adapt fast enough to delay spikes. According to Ramjee et al (1994) this algorithm success to increase their delay estimate on detection of a spike and again to decrease their estimate once the spike is over.

The delay variation estimate remains the same in the four algorithms and is calculated as:

$$\hat{v}_i = \hat{v}_{i-1} + (1-\alpha)\left|\hat{d}_i - n_i\right| \tag{3}$$

where n_i is the network delay experienced by packet *i* and α is chosen according to the selected algorithm (see Figure 1).

Several research studies have been devoted to find a better estimate for the network delay [DeLeon and Sreenan 1999], [Sreenan 2000], [Narbutt, and Murphy 2004]. However, little work has been done with respect to the impact of the Ω coefficient in the performance of the algorithms. In this paper we propose a simple mechanism to dynamically adapt the value of Ω.

2.2 Statistical Network Transmission Model

Recommendation ITU-T P.1050 [ITU-T 2007] specifies an IP network model and scenarios for evaluating and comparing communications equipment connected over a converged wide area network. The test scenarios combine LAN, access and core network elements in a realistic way to create layer 3 IP network impairments that cause packets to experience varying delay or loss. In our work we use this recommendation to generate audio packet traces.

2.3 Performance assessment

Mean opinion score has been traditionally used to measure subjective perception of voice communication. MOS is given on a scale of 1-5, where a higher value corresponds to better quality. Since MOS is a subjective test difficult to be carried out in practical situations, some other objective tests have been developed. ITU recommendation G.107 describes the E-model [ITU-T 2003], a computational algorithm that incorporates impairment factors present in modern transmission networks. The output of the E-model is a scalar quality rating value, R, which is computed as:

$$R = Ro - Is - Id - Ie\text{-}eff + A \tag{4}$$

$$\mathrm{MOS}_{\mathrm{CQ}} = \begin{cases} 1 & R < 0 \\ 1 + 0.035R + R(R-60)(100-R)7\cdot 10^{-6} & 0 < R < 100 \\ 4.5 & R > 100 \end{cases} \tag{5}$$

where *Ro* represents, in principle, the basic signal-to-noise ratio, *Is* is a combination of all impairments which occur more or less simultaneously with the voice signal, *Id* represents the impairments caused by delay, *Ie-eff* represents impairments caused by low bit-rate codecs and also includes impairment due to packet losses, and *A* is the advantage factor. An estimated Mean Opinion Score for the conversational situation in the scale 1-5 can be obtained from the *R*-factor using equation (5). Table 1 shows the relation between *R*-value, MOS and user satisfaction.

Table 1. Equivalent R values into estimate MOS

User satisfaction	R factor (lower limit)	MOS (lower limit)
very satisfied	90	4.34
satisfied	80	4.03
some user dissatisfied	70	3.60
many user dissatisfied	60	3.10
nearly all user dissatisfied	50	2.58

3. Classical Algorithms Performance

3.1 Experiments Setup

The performance of the four playout buffer algorithms is assessed using statistical models. The network scenario used in the evaluation process is depicted in Figure 2. Eight packet traces were generated with different network impairment severities according to [ITU-T 2007] (see Table 2 for details). The size of the audio packets was fixed to 200 bytes (160 byte corresponding to 20 ms of audio plus 40 byte of protocols overhead) to simulate a G.711 codec. Voice activity detection was implemented by means of speech models provided in [ITU-T 1993].

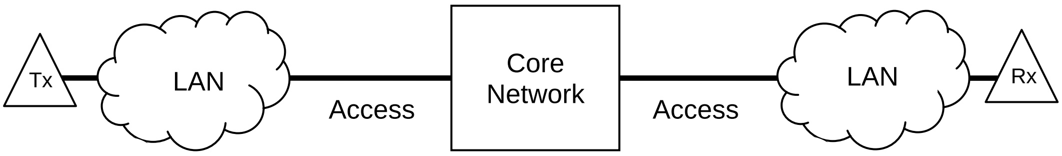

Figure 2. Network diagram.

Table 2. Trace details

Trace	1	2	3	4	5	6	7	8
Impairment								
LAN occupancy (%)	1	2	3	5	8	12	16	20
Access occupancy (%)	0	1	2	4	8	15	30	50
MTU (bytes)	512	512	1508	1508	1508	1508	1508	1508
Route flap interval (s)	0	3600	1800	900	480	240	120	60
Route flap delay (ms)	0	2	4	8	16	32	64	128
Core Delay (ms)	16	32	64	128	196	256	512	768
Jitter (ms)	5	10	24	40	70	100	150	500
Link fail interval (s)	0	3600	1800	900	480	240	120	60
Link fail duration (ms)	0	64	128	256	400	800	1600	3000
Packet loss (%)	0	0.01	0.02	0.04	0.1	0.2	0.5	1
Reordered packets (%)	0	0.00025	0.0005	0.001	0.005	0.01	0.05	0.1

Audio packet traces were passed through the playout buffer to estimate the mean mouth-to-ear delay and the overall packet loss rate. Finally, using these two parameters we compute the mean opinion score for conversational quality.

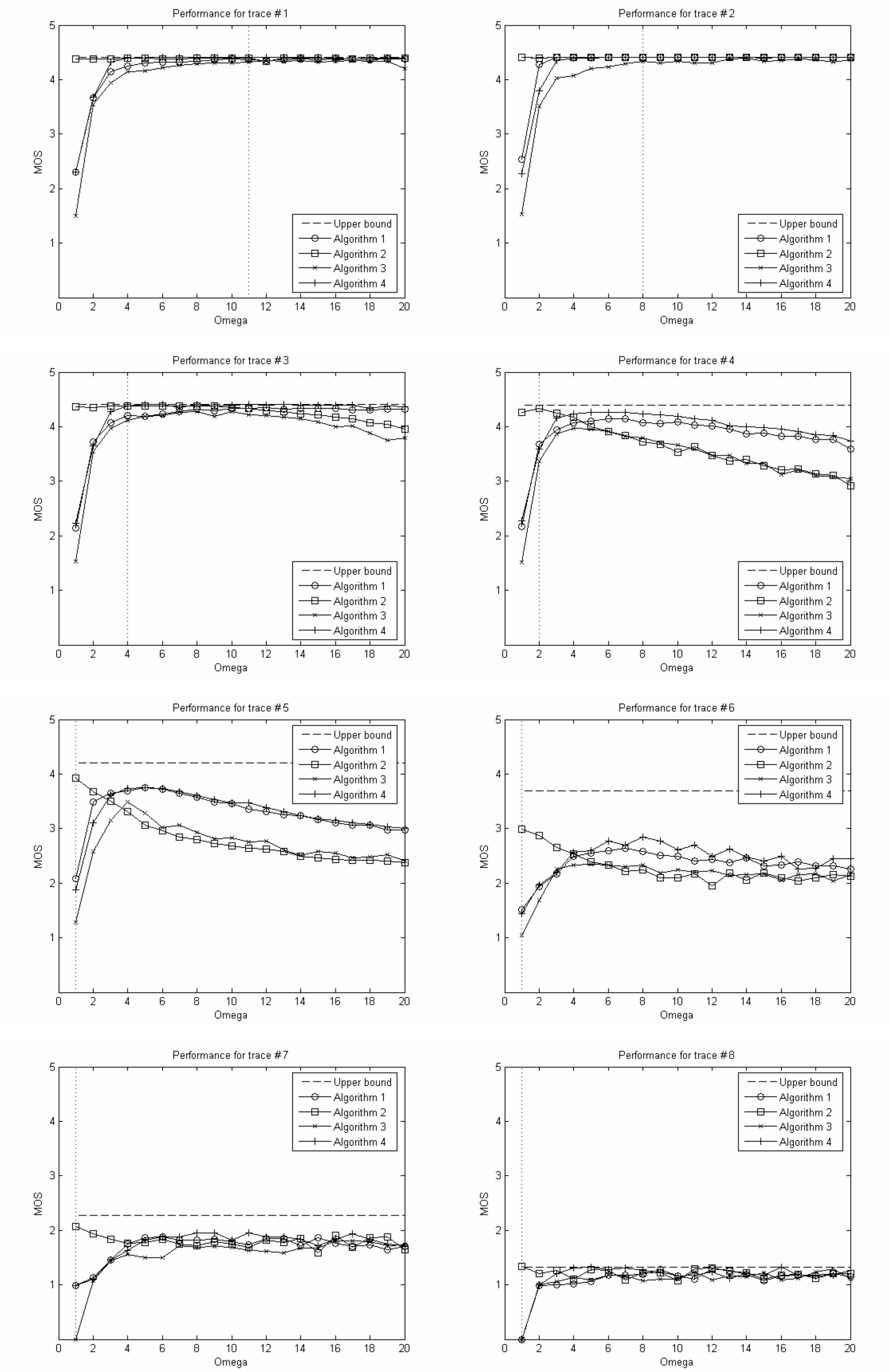

Figure 3. Algorithm performance varying the Ω (Omega) coefficient.

3.2 Algorithms comparison and discussion

In this section we compare the four classical playout buffer algorithms by varying the Ω coefficient. The MOS values obtained are shown in Figure 3. The horizontal dashed line corresponds to the upper bound of the MOS which is computed using the mean network delay and packet loss rate just before the playout buffer. Obviously for each of the four algorithms the audio quality degrades as the network impairment severities increase. However, the algorithms behave in different ways with respect to Ω; in several cases Ω = 4 do not produce the best result. Particularly Algorithm 2, for specific values of Ω, always achieves the best mean opinion score. These values are marked using a vertical dotted line.

The experiments reveal two interesting results in the case of Algorithm 2. The first one and more evident is that set Ω to 4 is not the best option for most of the sample audio traces. The second one is less evident and can be observed in Figure 4. In this figure we plot the values of Ω that maximize the MOS as a function of the mean network delay estimated by the algorithm. Now results quite obvious that exists an inverse relationship between the best value of Ω for the algorithm and the mean network delay. The Ω value decreases as the end-to-end network delay increases (i.e. when network impairment severities increase). This can be explained by the fact that Algorithm 2 adapts more quickly to burst of packet incurring in long delays. For that reason it tends to over estimate the mean network delay (see Figure 5). When a fixed amount of the jitter estimate is added to set the playout time the mouth-to-ear delay significantly increase.

Therefore, as the network delay increases it is possible to reduce the quantity, Ω, of jitter estimate added to the playout buffer without incurring in a noticeable number of packets out of time. Note that an increase in both, the mouth-to-ear delay and the packet loss rate, degrade the quality of service.

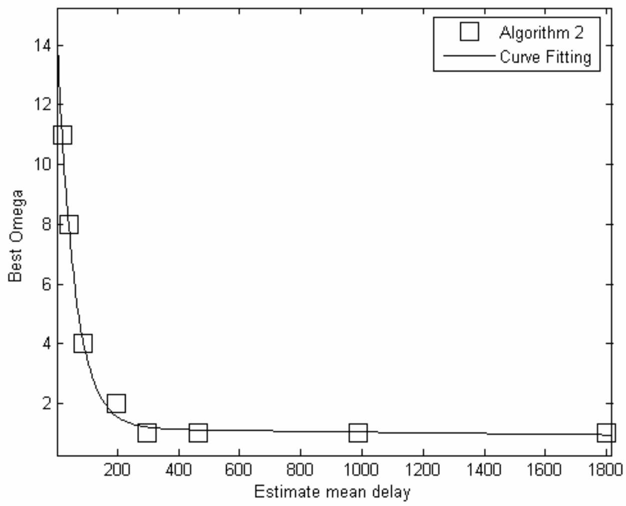

Figure 4. Best Ω (Omega) as a function of the estimate mean delay for Algorithm 2.

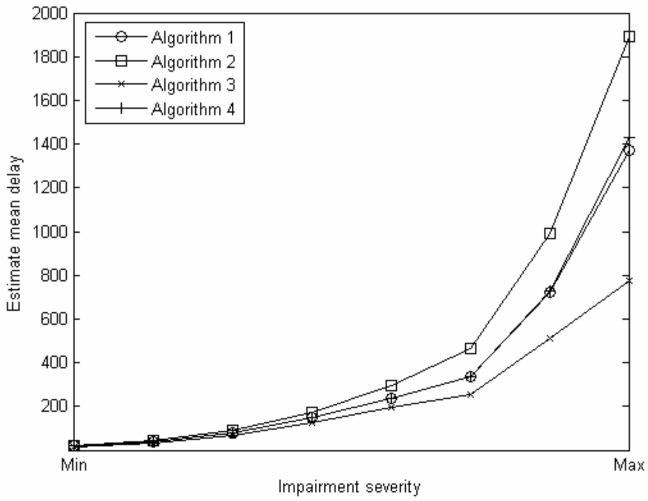

Figure 5. Estimate mean delay increasing the network impairment severities.

n_i : network delay for packet i

t_i : transmission time stamp of packet i

For the first packet of a talkspurt

Ω = min{max{$a/\hat{d}_i$,1},10}

$p_0 = \hat{d}_i + \Omega\hat{v}_i$

For every received packet

if $n_i > \hat{d}_i$

$\hat{d}_i = \beta\hat{d}_i + (1-\beta)n_i$

else

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

end

$\hat{v}_i = \alpha\hat{v}_i + (1-\alpha)\,|(\hat{d}_i - n_i)|$

$p_i = p_0 + t_i$

Figure 6. Algorithm 5.

4. Algorithm 5

Based on the above observations we propose a new algorithm, Algorithm 5, which dynamically adapts Ω to achieve better quality. Here we exploited the regularity found in Algorithm 2 (see Figure 4). The pseudocode for the algorithm is shown is Figure 6. The value of Ω is computed for the first packet of a talkspurt as

$$\Omega = \min\left(\max\left(\frac{a}{\hat{d}_i},1\right),10\right) \tag{6}$$

where a is constant and was set to 200 in our experimental studies. This value was chosen to best fitting a large data set collected in various network scenarios with different network impairment severities.

The idea is to reduce the values of Ω as the estimate network delay increases. The estimation of the network delay is the same as in Algorithm 2. The coefficient values are bounded between 1 and 10 since values out of that range lacks of practical use (e.g. $\Omega \to \infty$ as $\hat{d}_i \to 0$) and has not a significant impact in the performance. A large value of Ω for short network delays not only provides the higher MOS score but also permits to adsorb sudden and large increases in the end-to-end network delay (e.g. spikes). As soon as the network delay remains high the value of Ω move toward 1 in order to compensate the over estimation introduced by Algorithm 2.

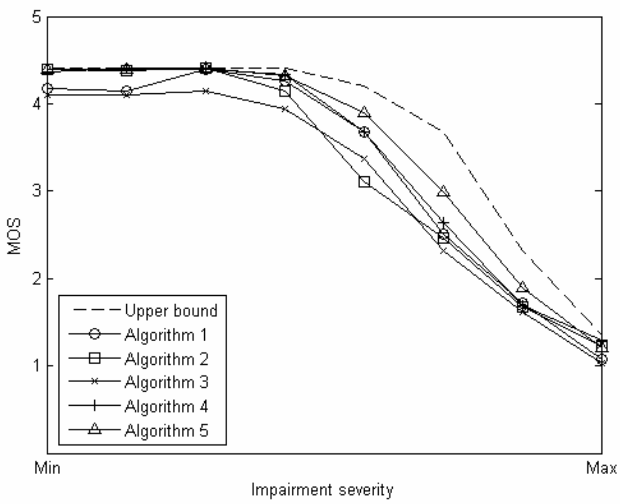

Figure 7. Comparison of the five algorithms under different network impairment severities.

A comparison between the original four algorithms (Ω = 4) and Algorithm 5 is shown in Figure 7. Algorithm 5 exhibits the best performance followed by Algorithm 4 and 1 in that order. This result is consistent with several studies that report Algorithm 4 to be the best ranking of the classical algorithms [Ramjee et al 1994], [Kansal and Karandikar 2001], [Narbutt et al 2005]. It is important to note that the computational cost of Algorithm 5 is comparable to that of Algorithm 2 and less heavy than Algorithm 4. The quality gain results significant with respect to the four classical algorithms. Specifically in the range of moderate to intense network impairment severities our algorithm clearly outperforms the original Ramjee's proposal.

5. Conclusions

In this paper we studied a classical playout buffer algorithm, exploring the effect of one of its parameters in the performance of the algorithm. This parameter has been traditionally skipped by the researches. Our results revealed an important unexploited regularity. Based on that knowledge we propose a new algorithm that achieves the highest score when compared to its classical predecessors. The proposed algorithm can bee carried out with minimal computational cost, resulting in a simple and easy way to implement a playout buffer mechanism for real multimedia applications.

References

Atzori, L and Lobina, M. L. (2006) “Playout Buffering IP Telephony: A Survey discussing Problems and Approaches” In IEEE Communications Surveys&Tutorials, pages 36-46

DeLeon P. and Sreenan, C. J. (1999) “An Adaptive Predictor for Media Playout Buffering,” In Proc. ICASSP, vol. 6, pages 3097–100.

ITU-T Recommendation P.59, “Artificial Conversational Speach ,” 1993

ITU-T Recommendation G.107, “The E-Model, a Computational Model for Use in Transmission Planning,” 03/2003.

ITU-T Recommendation G.1050, “"Network model for evaluating multimedia transmission performance over Internet Protocol",” 11/2007.

Kansal, A. and Karandikar A. (2001) “Jitter-Free Audio Playout Over Best Effort Packet Networks” In Proc ATM Forum, New Delhi, India.

Narbutt, M. and Murphy, L. (2004) “Improving Voice Over IP Subjective Call Quality,” In IEEE Commun. Letters, vol. 8, no. 5 pages 308–10.

Narbutt, M. Kelly, A. Murphy, L. and Philip, P. (2005) “Adaptative VoIP Playout Scheduling: Assesssing User Satisfaction “In IEEE Internet Computing, vol .9, no 4 , pages 28-34.

Ramjee, R. and Kurose, J. and Towsley D. and Schulzrinne H. (1994) “Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks,” In Proc. INFOCOM, Toronto, Canada.

Sreenan C. J. (2000) “Delay Reduction Techniques for Playout Buffering,” In IEEE Trans. Multimedia, vol. 2, no. 2 pages 88–100.

Jacobson V. (1988) “Congestion avoidance and control,” In Proc ACM SIGCOMM Conf, Stanford, pages 314-329.