Adaptive Voice Smoothing with Optimal Playback Delay Based on the ITU-T E-Model

Shyh-Fang Huang [1], Eric Hsiao-Kuang Wu [2], and Pao-Chi Chang [3]

[1] Department of Electronic Engineering, National Central University, Taiwan, hsf@vaplab.ee.ncu.edu.tw

[2] Department of Computer Science & Information Engineering, National Central University, Taiwan, hsiao@csie.ncu.edu.tw

[3] Department of Communications Engineering, National Central University, Taiwan, pcchang@ce.ncu.edu.tw

Abstract. Perceived voice quality is mainly affected by IP network impairments such as delay, jitter and packet loss. Adaptive smoothing buffer at the receiving end can compensate for the effects of jitter based on a tradeoff between delay and loss to archive a best voice quality. This work formulates an online loss model which incorporates buffer sizes and applies the ITU-T E-model approach to optimize the delay-loss problem. Distinct from the other optimal smoothers, the proposed optimal smoother suitable for most of codecs carries the lowest complexity. Since the adaptive smoothing scheme introduces variable playback delays, the buffer re-synchronization between the capture and the playback becomes essential. This work also presents a buffer re-synchronization algorithm based on silence skipping to prevent unacceptable increase in the buffer preloading delay and even buffer overflow. Simulation experiments validate that the proposed adaptive smoother archives significant improvement in the voice quality.

1 Introduction

The rapid progress of the development of IP-base network has enabled numerous applications that deliver not only traditional data but also multimedia information in real time. The next generation network, like an ALL-IP network, is a future trend to integrate all heterogeneous wired and wireless networks and provide seamless worldwide mobility. In an All-IP network, one revolution of the new generation Internet applications will realize VoIP services that people can talk freely around through the mobile-phones, the desktops and VoIP telephones at any time and place. Unfortunately, the IP-based networks do not guarantee the available bandwidth and assure the constant delay jitters (i.e., the delay variance) for real time applications. In other words, individual transmission delays for a given flow of packets in a network may be continuing to change caused by varying traffic load and differing routing paths due to congestions, so that the packet network delays for a continuous series of intervals (i.e.

talkspursts) at the receiver may not be the same (i.e. constant) as the sender. In addition, a packet delay may introduce by the signal hand-out or the difference of bandwidth transportation in wireless/fixed networks.

For delay sensitive applications, a dominating portion of packet losses might be likely due to delay constraint. A late packet that arrives after a delay threshold determined by playback time is treated as a lost packet. A tight delay threshold not only degrades the quality of playback but also reduces the effective bandwidth because a large fraction of delivered packets are dropped. In fact, delay and loss are normally not independent of each other. In order to reduce the loss impact, a number of applications utilize an adaptive smoothing technique in which buffers are adopted to reduce the quality damage caused by loss packets. However, a large buffer will introduce excessive end-to-end delay and deteriorate the multimedia quality in interactive real-time applications. Therefore, a tradeoff is required between increased packet loss and buffer delay to achieve satisfactory results for playout buffer algorithms.

In the past, the works on the degradation of the voice quality consider the effect of packet loss, but not that of packet delay. Within literature on predicting delays, the use of Pareto distribution in [1] is of computing the distribution parameters and rebuilding the new distribution to predict the next packet delay, and the use of neural network models to learn traffic behaviors [2]. The use of Pareto distribution or a neural network model requires relatively high complexity or a long learning period. Therefore, we consider the smoothers [3]-[9] which employ statistical network parameters related with the voice characteristic, i.e. loss, delay and talk-spurt that have significant influence to the voice quality. They detect delay spike in traffic and quickly calculate the required buffer size to keep the quality as good as possible.

The E-model is a computational model, standardized by ITU-T in G.107, G.109 and G.113 which uses the various transmission parameters to predict the subjective quality of packetized voice. Unfortunately, the E-model is complex to analyze in the optimization process. An alternative study is to apply a simplified E-model, first proposed by R. Cole and J. Rosenbluth [10], based upon observed transport measurements in the VoIP gateways and the transport paths. Authors indicated the simplified E-model method requires more pattern cases for traces to enhance the validation. Atzori and Lobina [11], and L. Sun and E. Ifeachor [12] proposed to utilize a simplified E-model that considers the loss and delay together to set a dijitter time, which is the optimal playout delay derived by a dynamic programming-based solution. However, the usability and the accuracy of a simplified E-model will be limited by non-typical traffic patterns.

For perceptual-based buffer optimization schemes for VoIP, voice quality is evaluated as a key metric since it represents user's perceived QoS. However, it requires an efficient and accurate objective way to optimize perceived voice quality. To consider the well-defined delay and loss impairments of the E-model, we employ a complete E-model for the quality optimization to obtain optimal perceived voice quality.

In a packet switching network, without a resynchronization scheme, a playback clock with a minor frequency error will eventually cause a buffer overflow or an underflow at the receiving end. The overflow packets are usually discarded due to the finite buffer size and the real-time requirement. This discontinuity caused by discarded packets might create an unpleasant effect to the playback quality because the lost packets could be the important part of the signals. This effect is more serious for audio signals than video signals because human ears are more sensitive to the continuity of sounds than human eyes.

The contributions of this paper are three-fold: (i) A new method optimizing voice quality for VoIP is easily applied to codecs which were well-defined in the ITU-T E-model. (ii) Different from the other optimal smoothers, our optimal smoother has the lowest complexity with $O(n)$. (iii) A feasible scheme is introduced to solve the buffer re-synchronization problem.

2 Related Work

The Spike Detection (SD) algorithm has been studied by many researchers [3]-[9]. A delay spike is defined as a sudden and significant increase of network delay in a short period often less than one round-trip. This algorithm adjusts the smoothing size, i.e. playback delay, at the beginning of each talk-spurt. The results of this algorithm are therefore compared to the results obtained herein.

The SD Algorithm in [3] estimates the playout time p_i of the first packet in a talk-spurt from the mean network delay d_i and the variance v_i for packet i as

$$p_i = t_i + d_i + \gamma v_i \tag{1}$$

where t_i represents the time at which packet i is generated at the sending host and γ is a constant factor used to set the playout time to be "far enough" beyond the delay estimate such that only a small fraction of the arriving packets could be lost due to late arrival . The value of $\gamma = 4$ is used in simulations [3]. The estimates are recomputed each time a packet arrives, but only applied when a new talk-spurt is initiated.

The mean network delay d_i and variance v_i are calculated based on a linear recursive filter characterized by the factors α and β .

$$\begin{cases} \mathit{If}\ n_i > d_{i\text{-}1} \Rightarrow \begin{cases} d_i = \beta d_{i-1} + (1-\beta) n_i \\ v_i = \beta v_{i-1} + (1-\beta)\left|d_{i-1} - n_i\right| \end{cases} & \mathit{(SPIKE\_MODE)} \\ \mathit{If}\ n_i \le d_{i\text{-}1} \Rightarrow \begin{cases} d_i = \alpha d_{i-1} + (1-\alpha) n_i \\ v_i = \alpha v_{i-1} + (1-\alpha)\left|d_{i-1} - n_i\right| \end{cases} & \end{cases} \tag{2}$$

where n_i is the total "delay" introduced by the network [3] and typical values of α and β are 0.998002 and 0.75 [1], respectively.

The decision to select α and β is based on the current delay condition. The condition $n_i > d_{i-1}$ represents network congestion (*SPIKE_MODE*) and the weight β is used to emphasize the current network delay. On the other hand, $n_i \le d_{i-1}$ represents network traffic is stable, and α is used to emphasize the long-term average.

In estimating the delay and variance, the SD Algorithm utilizes only two values, α and β, which are simple but may not be adequate, particularly when the traffic is unstable. For example, an under-estimated problem is when a network becomes spiked, but the delay n_i is just below the d_{i-1}, the SD Algorithm will evaluate the network to be stable and will not enter the *SIPKE_MODE.*

3 Adaptive Smoother with Optimal Delay-Loss Trade off

The proposed optimal smoother is derived using the E-model to trade off the delay and loss. This method involves, first, building the traffic delay model and the loss model. Second, the delay and loss impairments of the E-model are calculated according to the delay and the loss models. Third, the E-model rank R is maximized and thus the delay and loss optimized solution is obtained.

3.1 E-model Description

In the E-model, a rating factor *R* represents voice quality and considers relevant transmission parameters for the considered connection. It is defined in [13] as:

$$R = Ro - Is - Id - Ie\_eff + A \tag{3}$$

Ro denotes the basic signal-to-noise ratio, which is derived from the sum of different noise sources which contain circuit noise and room noise, send and receive loudness ratings. Is denotes the sum of all impairments associated with the voice signal, which is derived from the incorrect loudness level, non-optimum sidetone and quantizing distortion. Id represents the impairments due to delay of voice signals, that is the sum of Talker Echo delay (*Idte*), Listener Echo delay (*Idle*) and end-to-end delay (*Idd*). Ie_eff denotes the equipment impairments, depending on the low bit rate codecs (*Ie, Bpl*) and packet loss (*Ppl*) levels. Finally, the advantage factor *A* is no relation to all other transmission parameters. The use of factor *A* in a specific application is left to the designer's decision.

3.2 The Delay and Loss Models in E-model

For perceived buffer design, it is critical to understand the delay distribution modeling as it is directly related to buffer loss. The characteristics of packet transmission delay over Internet can be represented by statistical models which follow Exponential distribution for Internet packets (for an UDP traffic) has been shown to consistent with an Exponential distribution [14]. In order to derive an online loss model, the packet end-to-end delay is assumed as an exponential distribution with parameter $1/\mu$ at the receiving end for low complexity and easy implementation. The CDF of the delay distribution $F(t)$ can also be represented by [15]

$$F(t) = 1 - e^{u^{-1}t} \tag{4}$$

and the PDF of the delay distribution $f(t)$ is

$$f(t) = \frac{dF(t)}{dt} = \mu^{-1} e^{-\mu^{-1}t} \tag{5}$$

where μ is defined as the inverse of the average mean delay.

In a real-time application, a packet loss that is solely caused by extra delay can be derived from the delay model $f(t)$. The value of t_b represents the smoothing time for a smoother. When a packet delay exceeds t_b, a packet loss will occur. The loss function $l(t_b)$ can be derived as

$$l(t_b) = \int_{t_b}^{\infty} f(t)dt = \left(-e^{-\lambda t}\right)\Big|_{t_b}^{\infty} = -e^{-\infty} + e^{-\mu^{-1}t_b} = e^{-\mu^{-1}t_b} \tag{6}$$

3.3 Optimization on E-model

The delay and loss factors over transmission have greater impacts to the voice quality than the environments or equipments. To simplify the optimization complexity, and investigate on delay and loss impairments, we make three assumptions in a communication connection as the following: (i). The circuit noise, room noise and terminate signals will not change. (Ro and Is are fixed). (ii). An echo delay in the Sender/Receiver will not change. (Idte and Idle are fixed). (iii). A codec will not change (*Ie* is fixed). In [13], *R* is rewritten as Eq. (7)

$$R = (Ro - Is - Idte - Idle + A) - Idd - Ie\_eff \tag{7}$$

where *Idd* is approximated by

$$Idd = 25\left\{\left(1 + X^6\right)^{1/6} - 3\left(1 + \left[\frac{X}{3}\right]^6\right)^{1/6} + 2\right\},\ X = \frac{ln\left(\frac{t}{100}\right)}{ln(2)}, \tag{8}$$

when $T_a > 100$ ms and *Idd* =0 when $T_a \leq 100$,

and

$$Ie\_eff = Ie + (95 - Ie)\cdot\frac{Ppl}{Ppl + Bpl} \tag{9}$$

Factors *Ie* and *Bpl* are defined in [16] and T_a is one-way absolute delay for echo-free connections.

Due to the three assumptions above, the optimization process can be concentrated on the parameters of *Idd* and *Ie_eff* . Eq. (7) is derived to yield Eq. (10)

$$R = Constant - 25\left\{\left(1+X^6\right)^{1/6} - 3\left(1+\left[\frac{X}{3}\right]^6\right)^{1/6} + 2\right\} - (95 - Ie)\cdot\frac{Ppl}{Ppl+Bpl}, \tag{10}$$

$$\text{when } t > 100 \text{ ms}$$

The differential equation dR/dt is assigned to zero to maximize R to yield the best quality. According to Eq. (6), the loss probability $Ppl = e^{-\mu^{-1}t_b}$, so we can get

$$R' = 25*\left\{\left(1+X^6\right)^{\frac{-5}{6}} X^5 X' - \left[1+\left(\frac{X}{3}\right)^6\right]^{\frac{-5}{6}}\left(\frac{X}{3}\right)^5 X'\right\} - \frac{\mu^{-1}e^{-\mu^{-1}t}\cdot Bpl}{Bpl^2 + e^{-2\mu^{-1}t} + 2\cdot Bpl\cdot e^{-\mu^{-1}t}} = \tag{11}$$

$$, \; X = \frac{log\frac{t}{100}}{log\,2}, \; X' = \frac{1}{log\,2\cdot t}$$

The solutions for t are difficult to get directly from Eq. (11) since it contains the complex polynomial and exponential function, Therefore, we will try to solve the best smoothing time t with a numerical approach.

We notice the following three conditions. (i). In Eq. (8), when the smoothing time $t \le 100$ ms, *Idd* is zero (no delay impairment). It implies a smoother should set the minimum smoothing delay to 100 ms to prevent the most packet loss. (ii). The maximum end-to-end delay of 250ms is acceptable for most user applications to prevent serious voice quality destruction. (iii). For a common low bit rate codec, like G.723.1 and G.729, the frame rate is 30 ms and 10 ms, respectively, so the gcd(10,30) is 10 ms. Based on the above conditions, we can study the fifteen cases, $t_1 = 110\,ms$, $t_2 = 120\,ms$, …, $t_{15} = 250\,ms$, to calculate the correspondence, μ_1, μ_2, …, μ_{15}, by the numerical analysis in Eq. (11) and an error is less than 0.001. Table 1 shows the smoothing time t corresponding to μ . We can observe that as μ increases, the smoother will enlarge the smoothing time to smooth the late packets. According to Table 1, the proposed smoother will calculate the current μ ($\mu_{current}$) at the beginning of each talk-spurt and search for a minimum *n* which satisfies $\mu_n \ge \mu_{current}$. The optimal smoothing time will be $100 + n*10$ ms to keep the optimal voice quality.

Table 1. The relation of smoothing time and arrival rate

smoothing time	μ (1/sec)	smoothing time	μ (1/sec)
$t_1 = 110\,ms$	$\mu_1 = 9.71$	$t_9 = 190\,ms$	$\mu_9 = 16.95$
$t_2 = 120\,ms$	$\mu_2 = 10.64$	$t_{10} = 200\,ms$	$\mu_{10} = 17.86$
$t_3 = 130\,ms$	$\mu_3 = 11.49$	$t_{11} = 210\,ms$	$\mu_{11} = 18.52$
$t_4 = 140\,ms$	$\mu_4 = 12.35$	$t_{12} = 220\,ms$	$\mu_{12} = 19.61$
$t_5 = 150\,ms$	$\mu_5 = 13.33$	$t_{13} = 230\,ms$	$\mu_{13} = 20.41$
$t_6 = 160\,ms$	$\mu_6 = 14.08$	$t_{14} = 240\,ms$	$\mu_{14} = 21.28$
$t_7 = 170\,ms$	$\mu_7 = 14.93$	$t_{15} = 250\,ms$	$\mu_{15} = 22.22$
$t_8 = 180\,ms$	$\mu_8 = 15.87$		

4 Buffer Re-synchronization

A necessary condition that a smoother can work correctly is the synchronization between the capture and the playback. This section proposes a buffer re-synchronization machine (BRM) to help synchronization and the clock drift analysis of re-synchronization to validate the effectiveness.

4.1 Buffer Re-synchronization Machine

This work proposes a synchronization scheme that segments audio signals by detecting silences. The mismatch between the capture and the playback clocks is solved by skipping silences at the receiving end. The duration of the silent period may be shortened negligibly degrading the quality of playback. An active packet contains voice-compressed data, whereas a silent packet does not. Skipping some silent packets will not significantly degrade the quality of the voice, but can efficiently prevent the buffer from overflowing. Notably, k (could be adjusted) continuous silent packets could be utilized to separate different talkspurts.

Figure 1 depicts the buffer re-synchronization algorithm. Init-state, Smooth-state, Play-state and Skip-state are used to represent the voice conference initialing, the buffer smoothing, the buffer playing out, and the silent packets skipping, respectively, and "A" and "S" represents an active packet and a silent packet, respectively.

In the Init-state the buffer waits for the first arriving packets to initialize a voice conference. If Init-state receives an "S", it stays in Init-state; otherwise when an "A" is received, the Smooth-state is activated to smooth the packets. In the Smooth-state, the smoothing time b is computed by applying the optimal adaptive smoother algorithm dynamically. When the buffer smoothing time is over b, the Play-state is activated; otherwise it stays in Smooth-state for smoothing. In the Play-State the packet is fetched from the buffer and played out. In fetching process, when it encounters three consecutive S packets, implying that the talk-spurt can be ended, the buffer re-synchronization procedure then switches to the Skip-state. In the Skip-state, if "A" is fetched from buffer, it means the new talk-spurt has begun, and then it skips remained silent packets in the buffer, and switches to the Smooth-state to smooth the next talk-spurt. Otherwise, if "S" is fetched from buffer, it implies current talk-spurt is not ended and will be decoded to play out at the same state.

With the above four-state machine, the smoother can smooth the packets at the beginning of the talkspurt to avoid buffer underflow in the Smooth-state and skip the silent packets at the end of the talkspurt to prevent the overflow in the Skip-state.

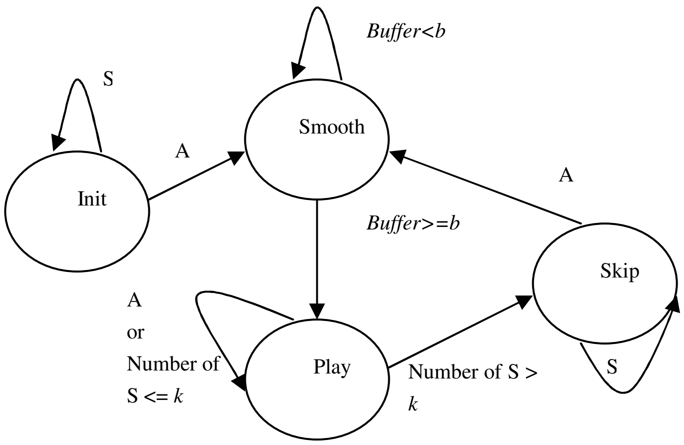


Fig. 1. Buffer Re-synchronization Machine

4.2 Effectiveness of Re-synchronization

To demonstrate the effectiveness of re-synchronization machine for buffer overflow, we analyze the clock inconsistence constraint as the following. C_s and C_r represent the sender clock (frame/sec) and the receiver clock, respectively, and M_a and M_s denote the mean active packets and mean silent packets in a talkspurt, respectively. The buffer overflow caused by the clock inconsistence (difference) will occur when C_s is larger than C_r condition. $C_s - C_r$, the difference value by subtracting the receiver clock from the sender clock, represents the positive clock drift between the sender and the receiver. Therefore, $(C_s - C_r) * ((M_a + M_s) * frame\_time)$ represents the mean extra buffer size caused by the positive clock drift for a mean talkspurt time. In order to distinguish the consecutive talkspurts, at lease k silent packets are utilized. Therefore, the smoother has $M_s - k$ silent packets to be skipped and resynchronizes with the following talkspurt. When the re-synchronization machine satisfies

$$(C_s - C_r) * ((M_a + M_s) * frame\_time) \le (M_s - k) \qquad \textbf{(12)}$$

, the buffer overflow caused by the positive clock drift will not occur.

5 Simulation

5.1 Simulation Configuration

A set of simulation experiments are performed to evaluate the effectiveness of the proposed adaptive smoothing scheme. The OPNET simulation tools are adopted to trace the voice traffic transported between two different LANs for a VoIP environment. Ninety personal computers with G.729 traffics are deployed in each LAN. The duration and frequency of the connection time of the personal computers follow Exponential distributions. Ten five-minute simulations were run to probe the backbone network delay patterns, which were used to trace the adaptive smoothers and compare the effects of the original with the adapted voice quality latter.

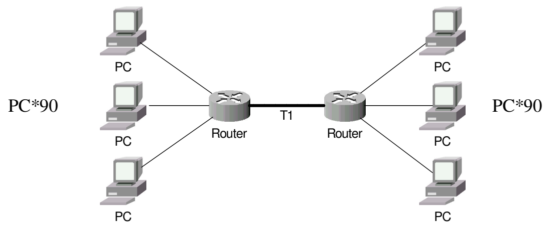
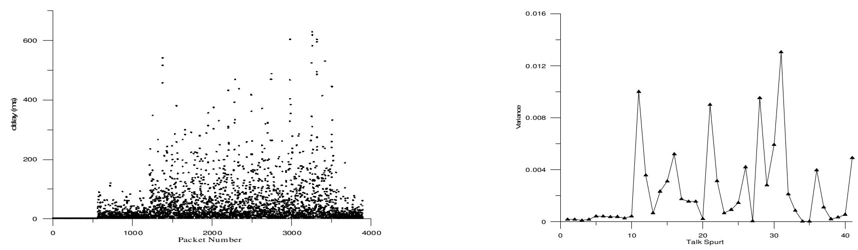


Fig. 2. The simulation environment of VoIP



(a) The delay of traffic

(b) The variance of traffic

Fig. 3. VoIP traffic pattern

Fig. 2 shows the typical network topology in which a T1 (1.544 Mbps) backbone connects two LANs, and 100 Mbps lines are connected within each LAN. The propagation delay of all links is assumed to be a constant value and will be ignored (the derivative value will be zero) in the optimization process. The buffer size of the bottlenecked router is assumed to be infinite since the performance comparison of adaptive smoothers will be affected by overdue packet loss (over the deadline) and not affected by the packet loss in router buffer. The network end-to-end delay of a G.729 packet with data frame size (10 bytes) and RTP/UDP/IP headers (40 bytes) is measured for ten five-minute simulations by employing the OPNET simulation network. Figure 3(a) and 4(b) list one of the end-to-end traffic delay patterns and the corresponding delay variances for VoIP traffic observed at a given receiver.

5.2 Voice Quality in smoothers

The test sequence is sampled at 8 kHz, 23.44 seconds long, and includes English and Mandarin sentences spoken by male and female. Fig. 4 shows the E-model score *R* of the voice quality. It shows that the optimal method has the significant improvement in the voice quality over SD smoother, because our proposed optimal smoother truly optimizes with the delay and loss impairments in a transmission planning of the E-model.

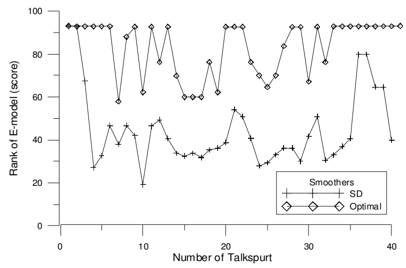


Fig. 4. The quality scores of smoothers

5.3 Re-synchronization Effectiveness for the Positive Clock Drift

A listening evaluation experiment was performed to analyze the required proper number of silent packets to segment the consecutive talk-spurts well. It was found in our experiments that at least three silent packets (e.q. 10 ms per packet in G.729) are required to separate talkspurts.

We analyze the G.729 voice sources used in our experiments and find the percentage of the mean active and mean silent segment length in a talkspurt are 0.51 and 0.49 respectively, and the maximum talkspurt length is 257 packets. p =3 is adopted to segment the consecutive talkspurt. From the Eq. (12), we can calculate the effective clock drift between the sender and the receiver $C_s - C_r$ should be less than or equal to $(257 * 0.49 - 3) \Big/ \left((257) * 10 * 10^{-3} \right) = 47.8$ (frame/sec). Normally, the clock drift will not be over 47.8 (frame/sec) when a sender of G.729 transmits 100 (frame/sec) to the networks. Consequently, the smoother can avoid the buffer overflow well in our case.

6 Conclusion

This article proposes an adaptive smoothing algorithm that utilizes the complete E-model to optimize the smoothing size to obtain the best voice quality. The buffer re-synchronization algorithm is also proposed to prevent buffer overflow by skipping some silent packets of the tail of talk-spurts. It can efficiently solve the mismatch between the capture and the playback clocks. Numerical results have shown that our proposed method can get significant improvements in the voice quality which balances the target delay and loss.

References

[1]. Brazauskas V., Serfling R.: Robust and efficient estimation of the tail index of a one-parameter pareto distribution. North American Actuarial Journal available at http://www.utdallas.edu/~serfling. (2000)

[2]. Tien P. L., Yuang M. C.: Intelligent voice smoother for silence-suppressed voice over internet. IEEE JSAC, Vol. 17, No. 1. (1999) 29-41

[3]. Ramjee R., Kurise J., Towsley D., Schulzrinne H.: Adaptive playout mechanisms for packetized audio applications in wide-area networks. Proc. IEEE INFOCOM. (1994) 680-686

[4]. Jeske D. R., Matragi W., Samadi B.: Adaptive play-out algorithms for voice packets. Proc. IEEE Conf. on Commun., Vol. 3. (2001) 775-779

[5]. Pinto J., Christensen K. J.: An algorithm for playout of packet voice based on adaptive adjustment of talkspurt silence periods. Proc. IEEE Conf. on Local Computer Networks. (1999) 224-231

[6]. Liang Y. J., Farber N., Girod B.,: Adaptive playout scheduling using time-scale modification in packet voice communications. Proc. IEEE Conf. on Acoustics, Speech, and Signal Processing, Vol. 3. (2001) 1445-1448

[7]. Kansal A., Karandikar A.: Adaptive delay estimation for low jitter audio over Internet. IEEE GLOBECOM, Vol. 4. (2001) 2591-2595

[8]. Anandakumar A. K., McCree A., Paksoy E.: An adaptive voice playout method for VOP applications. IEEE GLOBECOM, Vol. 3. (2001) 1637-1640

[9]. DeLeon P., Sreenan C. J.: An Adaptive predictor for media playout buffering. Proc. IEEE Conf. on Acoustics, Speech, and Signal Processing, Vol. 6. (1999) 3097-3100

[10]. Cole R., Rosenbluth J.: Voice over IP performance monitoring, Journal on Computer Commun. Review, Vol. 31. (2001)

[11]. Atzori L., Lobina M.: Speech playout buffering based on a simplified version of the ITU-T E-model. IEEE Signal Processing Letters, Vol. 11 , Iss 3. (2004) 382-385

[12]. Sun L., Ifeachor E.: New models for perceived voice quality prediction and their applications in playout buffer optimization for VoIP networks. Proc. ICC. (2004)

[13]. ITU-T Recommendation G.107,: The E-model, a Computational Model for use in Transmission Planning. (2003)

[14]. Bolot J. C.: Characterizing end-to-end packet delay and loss in the internet. Journal of High-Speed Networks, Vol. 2. (1993) 305-323.

[15]. Fujimoto K., Ata S., Murata M.: Statistical Analysis of Packet Delays in the Internet and its Application to Playout Control for Streaming Applications. IEICE Trans. Commun., Vol. E84-B, No. 6. (2001) 1504-1512

[16]. ITU-T SG12 D.106: Estimates of Ie and Bpl for a range of Codecs. (2003)