



# Perceptual-Based Playout Mechanisms for Multi-Stream Voice over IP Networks

*Chun-Feng Wu, Cheng-Lung Lee, and Wen-Whei Chang*

Department of Communications Engineering, National Chiao-Tung University, Hsinchu, Taiwan

wchang@cc.nctu.edu.tw

## Abstract

Packet loss and delay are two essential problems to real-time voice transmission over best-effort packet networks. In the proposed system, multiple descriptions of the speech are transmitted to take advantage of largely uncorrelated delay and loss characteristics on different network paths. Adaptive playout scheduling of multiple voice streams is formulated as an optimization problem leading to a better delay-loss tradeoff. Also proposed is a perceptually motivated optimization criterion based on a simplified version of the ITU-T E-model. Experimental results show that the proposed playout buffer algorithm improves the delay-loss tradeoff as well as speech reconstruction quality.

## 1. Introduction

Quality of service (QoS) has been one of the major concerns in the context of real-time voice communication over the Internet. Interactive audio applications such as telephony and audio conferencing require high constraints on packet loss and end-to-end delay. There has been much interest in the use of packet-level forward error correction (FEC) to compensate for loss, based on parity codes and Reed Solomon codes. All of the FEC mechanisms send some redundant information which is based on previously transmitted packets. Waiting for the redundant information results in a delay penalty, and consequently an increase in end-to-end delay. Previous research [1][2] proposed the use of multiple description coding (MDC) to exploit the largely uncorrelated delay and loss characteristics on different network paths. In MDC coders, the source is encoded into multiple descriptions that are then separately transmitted over independent network paths. Each description can be individually decoded for a reduced quality reconstruction of the source, but if all descriptions are available they can be jointly decoded for a higher quality reconstruction.

For the multi-stream voice transmission, the network delay experienced may vary for each packet depending on the paths taken by different streams and on the level of congestion along the path. Packets could get lost due to their late arrival resulting from excessive network delays. The variation in network delay, referred to as jitter, must be smoothed out since it obstructs the proper and timely reconstruction of the speech signal at the receiver end. The most common approach is to store recently arrived packets in a jitter buffer before playing them out at scheduled intervals. By increasing the buffer size, the number of late packet loss can be reduced at the cost of increased end-to-end delay. Thus, there is a need to develop playout buffer algorithms for a better balance between the end-to-end delay and packet loss. Most of the proposed playout buffer algorithms [1-4] focused on the delay-loss performance, but not the quality

perceived by end users. From the QoS perspective, these approaches are inappropriate as they do not provide a direct link to the perceived speech quality. Recent work [5-7] has considered new models for perceived voice quality prediction and their applications in playout buffer optimization for single-stream voice transmission. However, these approaches adjust the playout delay on a per-talkspurt basis and hence cannot be used in conjunction with per-packet delay adjustment algorithms. In this work, we will extend the concept of perceptual optimization to multi-stream voice transmission and propose a more general buffer design that can be applied to per-talkspurt as well as per-packet delay adjustment.

## 2. System Implementation

A block diagram of the proposed multi-stream voice transmission system is shown in Fig. 1. The system has five major components: MDC speech coder, Internet traffic simulator, delay distribution modelling, adaptive playout buffer, and time-scale modification. In the MDC coder, the speech signal is encoded into distinct descriptions for each of  $M$  channels, with the hope that at least one of the descriptions can be received correctly so that an acceptable quality of reconstructed speech can be achieved. Only the case with  $M = 2$  descriptions will be considered in this work, but the proposed technique can easily be extended to handle more descriptions. The best-effort nature of the Internet results in packets experiencing varying network delay due to different levels of congestion at the routers. To characterize this, we adopted the first-in-first-out (FIFO) queuing model described in [8] to simulate the network delay behavior of voice packets under a certain Internet workload. This queuing system can be viewed as a statistical multiplexer of voice stream and Internet stream. The voice stream is modelled by fixed-size packets arriving at regular intervals. The Internet stream is modelled as a mix of bulk traffic with larger packet size and interactive traffic with smaller packet size. The inter-arrival time for Internet packets is assumed to be exponentially distributed. With this model, we can easily generate different categories of network delay traces for performance evaluation of various playout buffer algorithms.

Delay jitter can be removed by buffering the received packets for a short period of time before playing them out at scheduled intervals. The playout delay of packet  $i$  is denoted by  $d_{play,i} = t_{p,i} - t_{s,i}$ , where  $t_{s,i}$  and  $t_{p,i}$  represent the time when the  $i$ th packet is sent and played out, respectively. Before the arrival of packet  $i$ , we have to determine the playout time for that packet according to the most recent delays we recorded. This task is accomplished by using an adaptive playout buffer algorithm that achieves the optimum perceived voice quality in the presence of jitter. To proceed with this, it is important to establish the delay distribution model as it is directly related to

10.21437/Interspeech.2007-471

late packet loss rate. Previous work in [7] has found that the delay characteristics of voice over Internet is better characterized by a Pareto distribution than a normal or an exponential distribution. For packet-based transmission, speech is usually processed and packetized into fixed size blocks and outgoing packets are generated at regular intervals with a constant period  $L_0$ , i.e.,  $t_{s,i+1} - t_{s,i} = L_0$ . When applying the per-packet delay adjustment, each individual voice packet may have a different playout delay and therefore its duration must be time-scaled such that the voice packet is played out just in time for the predicted playout time of the next packet. As a result, the length of speech segment that is played out for packet  $i$  is denoted by  $L_i = t_{p,i+1} - t_{p,i}$ . The time-varying factor,  $\rho_i = L_i/L_0$ , is then used to modify the time duration of packet  $i$ . The case of  $\rho_i > 1$  corresponds to a time-scale expansion, while the case of  $\rho_i < 1$  corresponds to a time-scale compression. In our earlier work [9], a time-scale modification technique was proposed based on a sinusoidal representation of the speech production mechanism.

### 3. Adaptive Multi-Stream Playout Framework

The main attraction of multi-stream voice transmission arises from its flexibility to trade off the end-to-end delay, losing both descriptions (packet erasure), and losing only one description. The latter two cases results in different degrees of speech quality degradation. Although there are methods which use fixed playout algorithms, better algorithms have been proposed that react to changing network conditions by dynamically adjusting the playout delay. The basic adaptive playout algorithm [1-4] operates by estimating two statistics characterizing the network delay, and uses them to calculate the playout delay as follows:

$$d_{play,i} = \hat{d}_i + \beta \hat{v}_i. \quad (1)$$

where  $\hat{d}_i$  and  $\hat{v}_i$  are running estimates of the mean and variance of network delay seen up to the  $i$ th arriving packet. Here  $\beta$  is the safety factor which can be used to set the playout time to be far enough beyond the delay estimate; so that only a small fraction of packets will arrive too late to be played out. A higher value of  $\beta$  results in a lower late loss rate as more packets arrive in time, however the end-to-end delay increases.

The strategies for estimating each packet's mean network delay can be divided into two steps. Firstly, we calculated the delay estimate  $\hat{d}_i^{(l)}$  for individual stream  $l$  ( $l = 1, 2$ ) based on its recorded past delays using the normalized least-mean-square (NLMS) algorithm [4]. Next, we used the values of  $\hat{d}_i^{(1)}$  and  $\hat{d}_i^{(2)}$  to determine the mean network delay  $\hat{d}_i$  for packet  $i$ . The NLMS algorithm is a linear adaptive filtering algorithm which aims to minimize the mean square error between the actual network delay  $n_i^{(l)}$  and its estimate  $\hat{d}_i^{(l)}$ . To proceed with this, the network delay of  $N = 18$  past packets in stream  $l$  is recorded and is denoted by  $\mathbf{n}_{i-1}^{(l)} = [n_{i-1}^{(l)}, n_{i-2}^{(l)}, \dots, n_{i-N}^{(l)}]^T$ . Past recorded delays are then passed through an FIR filter to compute the current estimate by

$$\hat{d}_i^{(l)} = \mathbf{h}_i^{(l)T} \mathbf{n}_{i-1}^{(l)}, \quad (2)$$

where  $\mathbf{h}_i^{(l)}$  is the tap-weight vector. The tap weights are updated using the following recursion

$$\mathbf{h}_{i+1}^{(l)} = \mathbf{h}_i^{(l)} + \frac{\mu}{\mathbf{n}_{i-1}^{(l)T} \mathbf{n}_{i-1}^{(l)} + b} \mathbf{n}_{i-1}^{(l)} \epsilon_i^{(l)}, \quad (3)$$

where  $\mu = 0.01$  is the step size,  $b$  is a small constant, and the estimation error is  $\epsilon_i^{(l)} = n_i^{(l)} - \hat{d}_i^{(l)}$ .

The next issue to be addressed is how to determine the mean network delay  $\hat{d}_i$  from its two estimates  $\hat{d}_i^{(1)}$  and  $\hat{d}_i^{(2)}$ . There are many possible variations on the estimation process, depending on relative emphasis placed on the delay and speech reconstruction quality. For this investigation, we chose to work with an estimate  $\hat{d}_i = \hat{d}_i^{(l^*)}$ , where  $l^* = \arg \min\{\hat{d}_i^{(l)}, l = 1, 2\}$ . We then applied an autoregressive approach to estimate the delay variance as  $\hat{v}_i = \alpha \hat{v}_{i-1} + (1 - \alpha)|n_i^{(l^*)} - \hat{d}_i|$ , where  $\alpha$  is a weighting factor used to control the convergence rate of the algorithm. Notice that the mean delay estimate  $\hat{d}_i$  is determined mainly by the first description arriving from either stream, suggesting that lower latency is given more emphasis than good reconstruction quality. In practice, this design strategy is desirable since the human perception is more sensitive to high latency, while increased quantization noise resulting from losing one description are less likely to be perceived as an impairment.

### 4. Perceptual optimization of playout delay

The safety factor  $\beta$  in equation (1) has a critical impact on the adjustment of playout delay, which in turn influences the delay-loss tradeoff. Compared with fixed  $\beta$  in existing playout algorithms [1-4], further enhancement is expected with dynamic setting of  $\beta_i$  for every packet  $i$ . In this work,  $\beta_i$  is adapted according to the observed delay distribution and the adopted criterion relies on the use of a simplified version of the conversational-quality E-model. The E-model, defined in the ITU-T Recommendation G.107 [10], is an analytic model of voice quality used for network planning purposes. It combines individual impairments due to both the signal's properties and the network characteristics into a single R-factor ranging from 0 to 100. In VoIP applications [11], the R-factor may be simplified as follows:  $R = 94.2 - I_d - I_e$ , where  $I_e$  is the equipment impairment factor and  $I_d$  is the delay impairment factor. The R-factor is related to the conversational mean opinion score ( $MOS_c$ ) through a fixed mapping in [10]. The derived  $I_e$  model has the following form:  $I_e = \gamma_1 + \gamma_2 \ln(1 + \gamma_3 e)$ , where  $e$  is the total loss rate and fitting parameters  $\gamma_i$ 's are codec-dependent. Similarly, the delay impairment factor can be derived by a simplified fitting process in the form

$$I_d = 0.024d + 0.11(d - 177.3)H(d - 177.3) \quad (4)$$

where  $d$  is the end-to-end delay and  $H(x)$  is the step function.

Perceptual-based buffer design must take into account the tradeoff between delay, packet loss, and speech reconstruction quality. We formulated this tradeoff as an optimization problem which involves finding the best value of the decision variable  $\beta_i$  for every packet  $i$ . By the best variable we mean the one that results in smallest value of the utility function defined by

$$I_{m,i}(\beta_i) = 0.024d_i + 0.11(d_i - 177.3)H(d_i - 177.3) + \gamma_1 + \gamma_2 \ln(1 + \gamma_3 e_i). \quad (5)$$

Here  $d_i$  represents the end-to-end delay of packet  $i$  which is a summation of the encoding delay  $d_c$  and the playout delay  $d_{play,i} = \hat{d}_i + \beta_i \hat{v}_i$ . That is  $d_i = d_c + \hat{d}_i + \beta_i \hat{v}_i$ . The loss probability of packet  $i$  is calculated as

$$e_i = e_n^{(1)} e_n^{(2)} + e_n^{(1)} (1 - e_n^{(2)}) e_{b,i}^{(2)} + e_n^{(2)} (1 - e_n^{(1)}) e_{b,i}^{(1)} + (1 - e_n^{(1)}) (1 - e_n^{(2)}) e_{b,i}^{(1)} e_{b,i}^{(2)} \quad (6)$$

where  $e_n^{(l)}$  and  $e_{b,i}^{(l)}$  represent the network loss probability and the late loss probability in stream  $l$ , respectively. Notice that  $e_{b,i}^{(l)}$  and  $d_{play,i}$  are strongly correlated, and to find out their relationship, the characteristics of network delay in stream  $l$  are assumed to follow a Pareto distribution which is defined as  $F_l(x) = 1 - (k_l/x)^{\alpha_l}$ . Pareto distribution parameters  $\{\alpha_l, k_l\}$  can be estimated from a network trace using the maximum likelihood estimation method [7]. Given a playout delay  $d_{play,i}$ , the late loss probability in stream  $l$  can then be calculated as  $e_{b,i}^{(l)} = 1 - F_l(d_{play,i}) = (k_l/d_{play,i})^{\alpha_l}$ .

Equation (5) is an expression of the utility function as a function of the safety factor  $\beta_i$ . By differentiating it with respect to  $\beta_i$ , we get the following equation for the gradient:

$$I'_{m,i}(\beta_i) = \frac{dI_{m,i}}{d\beta_i} = c\hat{v}_i + \frac{\gamma_2\gamma_3}{1 + \gamma_3e_i} \frac{de_i}{d\beta_i}. \quad (7)$$

where

$$c = \begin{cases} 0.024, & \beta_i < (177.3 - d_c - \hat{d}_i)/\hat{v}_i; \\ 0.134, & \beta_i > (177.3 - d_c - \hat{d}_i)/\hat{v}_i. \end{cases} \quad (8)$$

and

$$\frac{de_i}{d\beta_i} = \frac{-\hat{v}_i}{d_{play,i}} \{ (1 - e_n^{(1)})(1 - e_n^{(2)})e_{b,i}^{(1)}e_{b,i}^{(2)}(\alpha_1 + \alpha_2) + e_n^{(1)}(1 - e_n^{(2)})e_{b,i}^{(2)}\alpha_2 + e_n^{(2)}(1 - e_n^{(1)})e_{b,i}^{(1)}\alpha_1 \} \quad (9)$$

Proceeding in this way, the secant method [12] is then applied to find the perceptual optimum value of  $\beta_i$ . Starting with two initial values  $\beta_i(-1)$  and  $\beta_i(0)$ , the iterative formula for the secant algorithm has the form

$$\beta_i(j+1) = \beta_i(j) - \frac{\beta_i(j) - \beta_i(j-1)}{I'_{m,i}(\beta_i(j)) - I'_{m,i}(\beta_i(j-1))} I'_{m,i}(\beta_i(j)). \quad (10)$$

The new value  $\beta_i(j+1)$  is then used in the next iteration and the estimation process is repeated until the difference  $|\beta_i(j+1) - \beta_i(j)|$  is smaller than a threshold.

The proposed perceptual optimum playout buffer algorithm, when applied to multi-stream voice transmission, can be summarized as below.

1. Update network delay records for the past 200 packets in every stream  $l$  ( $l = 1, 2$ ), and use them to calculate the Pareto distribution parameters  $(\alpha_l, k_l)$  by the maximum likelihood estimation method.
2. Use the values of  $(\alpha_l, k_l)$  in the secant method to determine the optimal value of  $\beta_i$ .
3. Calculate the network delay estimate  $\hat{d}_i$  and the variance estimate  $\hat{v}_i$ . Then, set the playout delay to  $d_{play,i} = \hat{d}_i + \beta_i\hat{v}_i$ .

## 5. Experimental results

Experiments were carried out to investigate the potential advantages of using the perceptual-based multi-stream playout algorithm for voice communication over IP networks. Our efforts began with the simulated delay traces for use in two different voice streams. In stream 1, many spikes are observed in the delay trace and the mean delay and variance are 78.7 ms and 20.8, respectively. In stream 2, no dynamic changes in the network delays are observed and the mean delay and variance are 135 ms and 1.74, respectively. The network loss rate for stream

1 is chosen to be 2%, compared with 0.5% for stream 2. For low complexity, we use the MDC scheme described in [13] to generate two voice streams of equal importance at the sender. The procedure is to partition the source data into even-sample and odd-sample sets and then compress independently to produce two descriptions. The compressions is based on 12-bit PCM coding and the corresponding fitting parameters for the  $I_e$  model are  $\gamma_1 = 0$ ,  $\gamma_2 = 30$ , and  $\gamma_3 = 15$ . If one of these descriptions is lost, missing samples are estimated by linear interpolation between the two neighboring received samples.

We first compare our multi-stream transmission scheme with a scheme that uses a single description coder (SDC) based on 12-bit PCM coding. Performance metrics used to evaluate the schemes are the average playout delay and the late loss rate. The results are plotted in Fig. 2 for using stream 1 only, using stream 2 only, and using both MDC-coded streams. The continuous curves with different late loss rate and playout delay are obtained by varying the safety factor  $\beta$  in (1). From it we observe a significant reduction of the playout delay for a fixed target late loss rate when using the MDC scheme. At the same late loss rate of 5%, the MDC scheme yielded the lowest average delay of 103 ms, compared with 118 and 137 ms for SDC in stream 1 and stream 2, respectively. On the other hand, if fixing the same average playout delay, the MDC scheme also results in the lowest late loss rate. Our next experiment was conducted to determine whether the improved delay-loss tradeoff could also be realized perceptually. Table 1 compares the packet loss rate, average end-to-end delay, and  $MOS_c$  evaluated by simulation for various values of  $\beta$  (4, 6, and dynamic  $\beta$ ). Compared with SDC schemes, the better speech quality of resulting from the MDC scheme is clearly illustrated. Simulation results also show that the proposed  $\beta$ -adaptive scheme can enhance perceived speech quality for multi-stream voice transmission over IP networks.

## 6. Conclusions

In this paper, we proposed a perceptually motivated optimization criterion and a practically feasible new algorithm for multi-stream playout buffer design. We formulate the perceptual-based buffer design as an optimization problem leading to a better tradeoff between packet loss and end-to-end delay. We also compared the perceived speech quality using the E-model methodology for playout algorithms with fixed and dynamic setting of the safety factor. Experimental results show that the proposed multi-stream playout algorithm can achieve a better delay-loss tradeoff and thereby improves the perceived speech quality.

## 7. Acknowledgements

This study was jointly supported by MediaTek Inc. and National Science Council, Republic of China, under contract NSC 95-2221-E-009-078.

## 8. References

- [1] W. Jiang and A. Ortega, "Multiple description speech coding for robust communication over lossy packet networks," in *International Conference on Multimedia and Expo*, New York, USA, August . 2000, vol. 1, pp. 444–447.
- [2] Y.J. Liang, E.G. Steinbach, and B. Girod, "Multi-stream voice over IP using packet path diversity," in *Multimedia*

Signal Processing IEEE Fourth Workshop, 2001, pp. 555–560.

- [3] S.B. Moon, J. Kurose, and D. Towsley, “Packet audio playout delay adjustment: Performance bounds and algorithms,” *Multimedia Systems*, vol. 6, no. 1, pp. 17–28, Jan. 1998.
- [4] P. DeLeon, and C.J. Sreenan “An adaptive predictor for media playout buffering,” in *Proceedings ICASSP’99*, vol. 6, pp. 3097–3100, March. 1999.
- [5] L. Sun and E. Ifeachor, “New Models for Perceived Voice Quality Prediction and their Applications in Playout Buffer Optimization for VoIP Networks,” in *Proceedings of ICC 2004*, June 2004.
- [6] L. Atzori, and M.L. Lobina “Speech playout buffering based on a simplified version of the ITU-T E-Model,” *IEEE Signal Processing Letters*, June 2004.
- [7] K. Fujimoto, S. Ata, and M. Murata “Adaptive Playout Buffer Algorithm for Enhancing Perceived Quality of Streaming Applications,” in *Processings of IEEE Globecom2002*, Nov 2002.
- [8] J.C. Bolot, “Characterizing end-to-end packet delay and loss in the internet,” *Journal of High-Speed Networks*, vol. 2, pp. 305-323, Dec 1993
- [9] C.L. Lee, W.W. Chang, and Y.C. Chiang, “Spectral and prosodic transformations of hearing-impaired Mandarin speech,” *Speech Communication*, vol. 48, issue 22, pp 207-219, Feb. 2006.
- [10] International Telecommunication Union, “The E-model, a computational model for use in transmission planning,” *ITU-T Recommendation G.107*, July 2000.
- [11] R. Cole and J. Rosenbluth, “Voice over IP performance monitoring,” in *Journal on Computer Communication Review*, vol. 31, no. 2, Apr. 2001.
- [12] E.K.P. Chong and S.H. Zak, *An Introduction to Optimization*, John Wiley & Sons, Inc., 2001.
- [13] N.S. Jayant and S.W. Christensen, “Effects of packet losses in waveform coded speech and improvements due to an odd-even sample-interpolation procedure,” *IEEE Trans. Communications*, vol. COM-29, no. 2, Feb. 1981, pp.101-109.

Table 1: Performance comparison for different playout algorithm.

| Playout algorithms              | Delay(ms) | Loss(%) | MOSc |
|---------------------------------|-----------|---------|------|
| SDC (stream 1), dynamic $\beta$ | 142.15    | 2.03    | 4.13 |
| SDC (stream 2), dynamic $\beta$ | 168.87    | 0.56    | 4.28 |
| MDC, $\beta = 4$                | 109.17    | 4.00    | 3.93 |
| MDC, $\beta = 6$                | 114.39    | 1.07    | 4.26 |
| MDC, dynamic $\beta$            | 142.88    | 0.04    | 4.35 |

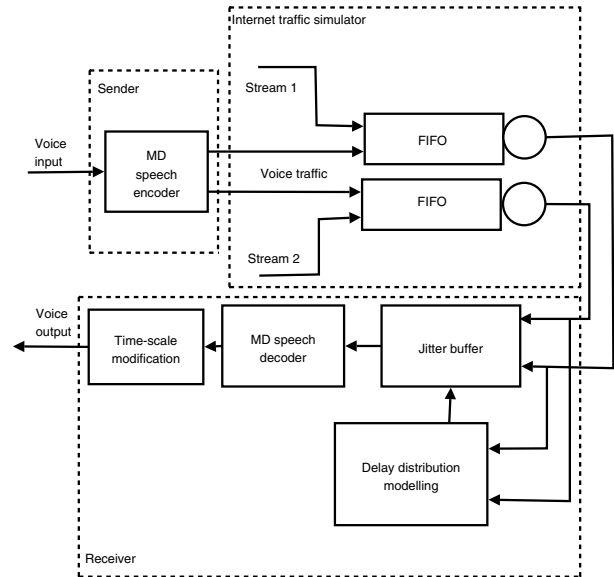


Figure 1: The multi-stream voice communication system diagram.

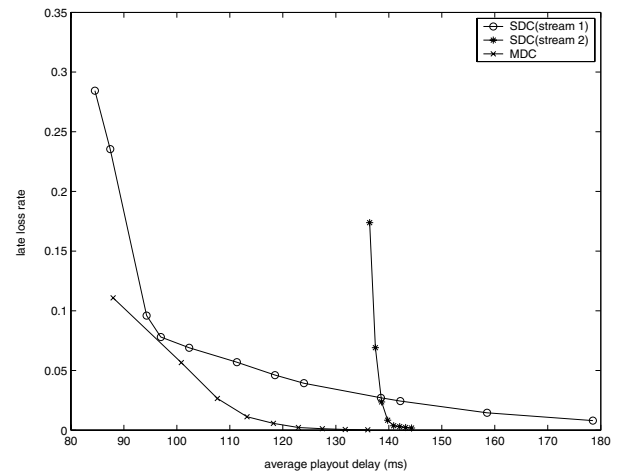


Figure 2: Delay-loss performance of adaptive playout algorithm tradeoff.