14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

ADAPTIVE PLAYOUT SCHEDULING FOR MULTI-STREAM VOICE OVER IP NETWORKS

Chun-Feng Wu, I-Te Lin and Wen-Whei Chang

Department of Communications Engineering, National Chiao-Tung University
Hsinchu, Taiwan
phone: + (886) 3-5731826, fax: + (886) 3-5710116, email: wwchang@cc.nctu.edu.tw

ABSTRACT

Packet delay and loss are two essential problems to real-time voice transmission over IP networks. In the proposed system, multiple redundant descriptions of the speech are transmitted to take advantage of largely uncorrelated delay and loss characteristics on independent network paths. Adaptive playout scheduling of multiple voice streams is then formulated as a constrained optimization problem leading to a better balance between end-to-end delay and packet loss. For proper reconstruction of continuous speech, we also develop a packet-based time-scale modification algorithm based on sinusoidal representation of the speech production mechanism. Experimental results indicate that the proposed adaptive multi-stream playout scheduling technique 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 voice communication over the Internet. Interactive audio applications such as telephony and audio conferencing require high constraints on packet loss, end-to-end delay and delay variation (jitter). 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. Recent research [1][2] proposed the use of multiple description (MD) coding to exploit the largely uncorrelated delay and loss behavior on multiple independent network paths. In MD coders, the source is encoded into multiple descriptions that are then separately transmitted over different 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. However, the network delay experienced may vary for each packet depending on the paths taken by different streams and on the level of congestion at the routers 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. A common method 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 scheduling algorithms for improved tradeoff between the end-to-end delay and packet loss. Previous work mainly focused on the delay concealment for single-stream transmission [3]-[6], except the work of Liang *et al.* [2]. Although there are methods which use fixed playout algorithms, adaptive algorithms have been proposed that react to changing network conditions by dynamically adjusting the playout delay. In conventional adaptive algorithms [3][4], the playout delay is adjusted on a per-talkspurt basis by lengthening or shortening the silence intervals between talkspurts. A better alternative [5][6] is per-packet based that performs the delay adjustment not only between talkspurts, but also within talkspurts. When applying such per-packet based algorithms, it is important to time-scale individual packets such that they are played out just in time for the predicted arrival time of the next packet. In this work, we propose an adaptive multi-stream playout scheduling algorithm that improves the delay-loss tradeoff as well as speech reconstruction quality. Also proposed is a packet-based time-scale modification algorithm based on a sinusoidal representation of sound production mechanism [7], as opposed to previous work based on the waveform similarity overlap-add (WSOLA) algorithm [8].

2. SYSTEM IMPLEMENTATION

A block diagram of the proposed multi-stream voice transmission system is shown in Figure 1. The system has four major components: MD speech coding, delay jitter model, adaptive playout scheduling, and time-scale modification. In the MD coder, the speech signal is encoded into two redundant descriptions, 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. For low complexity, we use the MD coding scheme described in [1] to generate two voice streams of equal importance at the sender. The basic idea is to quantize the even samples in finer resolution and the difference between adjacent even and odd samples in coarser resolution, and then packetize them into stream 1. For stream 2, we quantize even and odd samples in the opposite way. The redundancy of MD coding is 12.5% when using a 16-bit PCM coder as the fine quantizer and using a 2-bit ADPCM coder as the coarse quantizer. The two descriptions of the speech signal can be decoded independently at the receiver end. If both descriptions are available, the odd and even samples can be reconstructed in full resolution. When only the description of stream 1 is received, the even samples can be reconstructed in full resolution, while the odd samples are reconstructed at a coarser resolution. With this coding scheme, speech distortion resulting from losing one description only increases the quantization noise and is usually tolerated as a minor impairment.

The best-effort nature of the Internet results in packets experiencing varying network delay due to different levels of congestion in the network. To characterize this, we adopted the first-in-first-out (FIFO) queuing model described in [9] 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 interarrival time for Internet packets is assumed to be exponentially distributed. With this model, we can easily generate different categories of delay traces for performance evaluation of the proposed playout scheduling algorithm. 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 $d^i_{play} = t^i_p - t^i_s$, where t^i_s and t^i_p represent the time when packet i is sent and played out, respectively. Before the arrival of packet i, we have to determine the playout delay for that packet according to the most recent delays we recorded. This task is accomplished by using an adaptive multi-stream playout scheduling algorithm that improves the delay-loss tradeoff as well as speech reconstruction quality.

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., $t^{i+1}_s - t^i_s = L_0$. In this work, the playout delay adjustment is performed on a per-packet basis and therefore each individual voice packet may have a different scheduled playout time. For proper reconstruction of continuous speech, individual voice packets must be time-scaled such that they are played out just in time for the predicted playout time of the next packet. As a result, the length of audio segment that is played out for packet i is denoted by $L^i = t^{i+1}_p - t^i_p$. 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. A key issue in designing the time-scale modification system is to modify the time duration of a voice packet without changing its acoustic attributes. Our proposed time-scale modification algorithm is based on a sinusoidal representation of the speech production mechanism [7]. It consists of first analyzing the speech signal to obtain characteristic features, then applying the desired modification to these features, and synthesizing the corresponding signal. Notice that the proposed time-scaling technique is implemented at the receiver only, independently of the MD coding scheme used for transmission.

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. We formulate this tradeoff as a constrained optimization problem that involves finding a minimizer of the objective function $f(d^i_{play})$ over all possible values in the constraint set Ω^i. In our problem domain, the objective function that we wish to minimize is the network delay itself, i.e.,$f(d^i_{play}) = d^i_{play}$. Depending on relative emphasis placed on the delay and speech reconstruction quality, there are many possible variations on the constraint set. For this investigation, we chose to work with a constraint set which takes the form of functional constraints

$$\Omega^i = \{d^i_{play} : d^i_{play} \geq \hat{D}^i_{S_1} \bigcup d^i_{play} \geq \hat{D}^i_{S_2}\}, \qquad (1)$$

where $\hat{D}^i_{S_k}$ is the estimated end-to-end delay of packet i in stream k. The playout delay 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.

The dynamic setting of each packet's playout schedule is critical for the final performance of our multi-stream voice communication system. Since it involves switching between streams during speech playout, the end-to-end delay needs to be computed ahead of time for each individual stream based on its own past network delays. A typical approach is to estimate the end-to-end delay as

$$\hat{D}^i_{S_k} = \hat{d}^i_{S_k} + \beta \hat{v}^i_{S_k}, \qquad (2)$$

where $\hat{d}^i_{S_k}$ and $\hat{v}^i_{S_k}$ are estimates of the network delay and the jitter for packet i in stream k, respectively. The safety factor β is used to control the tradeoff between end-to-end delay and packet loss due to late arrival. A higher value of β results in a lower late loss rate as more packets arrives in time, however the end-to-end delay increases. The next issue to be addressed is how to estimate the network delay and the jitter. Here the network delay is estimated based on recorded past delays using the NLMS algorithm [6]. The NLMS algorithm aims to minimize the mean square error between the actual network delay $d^i_{S_k}$ and its estimate $\hat{d}^i_{S_k}$. The network delay of N past packets in each individual stream k is recorded and is denoted by $\mathbf{d}^i_{S_k} = [d^{i-1}_{S_k}, d^{i-2}_{S_k}, \ldots, d^{i-N}_{S_k}]^T$. Past recorded delays are then passed through an FIR filter to compute the current estimate by $\hat{d}^i_{S_k} = \mathbf{w}^{i\,T}_{S_k} \mathbf{d}^i_{S_k}$, where $\mathbf{w}^i_{S_k}$ is the $N \times 1$ vector containing the filter's tap weights. The tap weights of the filter are updated using the following recursion

$$\mathbf{w}^{i+1}_{S_k} = \mathbf{w}^i_{S_k} + \frac{\mu}{\mathbf{d}^{i\,T}_{S_k} \mathbf{d}^i_{S_k} + b} \mathbf{d}^i_{S_k} e^i_{S_k}, \qquad (3)$$

where μ is the step size, b is a small constant, and the estimation error $e^i_{S_k} = d^i_{S_k} - \hat{d}^i_{S_k}$. Given the network delay estimate for stream k, we then use an autoregressive approach to estimate the jitter as $\hat{v}^i_{S_k} = \alpha \hat{v}^{i-1}_{S_k} + (1-\alpha)|d^i_{S_k} - \hat{d}^i_{S_k}|$, where α is a weighting factor used to control the convergence rate of the algorithm.

4. PACKET-BASED TIME-SCALE MODIFICATION

When adjusting the playout schedule on a per-packet basis, it is important to maintain continuous playout by time-scaling individual packets without impairing speech quality. The proposed time-scale modification scheme is based on a sinusoidal representation of speech production mechanism [7].

Essentially, the production of sound can be described as the output of passing a glottal excitation signal through a vocal tract system. To track the nonstationary evolution of characteristic features, time-scale manipulations will be performed on a frame-by-frame basis. In this work, speech signals were analyzed using a 46.4 ms Hamming windows with a 18.75 ms frame shift. Therefore, the analysis frame interval Q was fixed at 18.75 ms. For the speech on the ith frame, the vocal tract system function can be described in terms of its amplitude function $A^i(w)$ and phase function $\Phi^i(w)$. Here the excitation signal is represented as a sum of M^i sine waves, each of which is associated with the frequency w_m^i and the phase Ω_m^i. A block diagram of the time-scale modification system is shown in Figure 2. The analysis begins by estimating from the Fourier transform of speech the fundamental frequency w_0^i, the voicing probability P_v^i, the amplitude function $A^i(w)$ and the phase function $\Phi^i(w)$ of the vocal tract system. The voicing probability will be used to control the harmonic spectrum cutoff frequency $w_c^i = \pi P_v^i$, below which the sine-wave frequencies $w_m^i = m w_0^i$ and above which $w_m^i = m w_0^i + w_u$, where w_u is the unvoiced term corresponding to 100 Hz. The system amplitudes M_m^i and phases Φ_m^i are then obtained by samples of their respective functions at the frequencies w_m^i.

With reference to the sinusoidal framework, the time-scale modification involves scaling the synthesis frame of original duration Q by a factor of ρ^i, i.e., $\hat{Q}^i = \rho^i Q$. Note that the time-scaling factor ρ^i is determined by the adaptive playout scheduler. After that, a two-step procedure is used in estimating the excitation phase $\hat{\Psi}_m^i$ of the mth sine wave. The first step is to obtain the onset time $\hat{n}_0^i$ relative to the new frame interval $\hat{Q}^i$. This is done by accumulating a succession of pitch periods until a pitch pulse crosses the center of the ith frame. The location of this pulse is the onset time $\hat{n}_0^i$ at which sine waves are in phase. The second step is to compute the excitation phase as follows:

$$\hat{\Psi}_m^i = -\hat{n}_0^i w_m^i + \varepsilon_m^i, \tag{4}$$

where the unvoiced phase component ε_m^i is zero for the case of $w_m^i \le w_c^i$ and is made random on $[-\pi, \pi]$ for the case of $w_m^i > w_c^i$. Finally, in the synthesizer the system amplitudes A_m^i are linearly interpolated over two consecutive frames. Also, the excitation and system phases are summed and the resulting sine-wave phases, $\hat{\theta}_m^i = \hat{\Psi}_m^i + \Phi_m^i$, are interpolated using the cubic polynomial interpolator. The final synthetic speech waveform on the ith frame is given by

$$s(n) = \sum_{m=1}^{M^i} A_m^i \cos[n w_m^i + \hat{\theta}_m^i],\ t_i \le n \le t_{i+1} - 1 \tag{5}$$

where $t_i = \sum_{j=1}^{i-1} \hat{Q}^i$ denotes the starting time of the current synthesis frame.

5. EXPERIMENTAL RESULTS

Experiments were carried out to investigate the potential advantages of using adaptive multi-stream playout scheduling for voice communication over IP networks. We fist used the FIFO queuing model to generate two categories of Internet traffic, one is light traffic used for stream 1 and the other is heavy traffic used for stream 2. In the heavy traffic case, some packets may pile up and result in consecutive large delays. For each stream, we send 18.75 ms UDP packets with payload size of 338 bytes, reflecting 16-bit PCM for finer quantization and 2-bit ADPCM for coarser quantization at 8 kHz sampling rate. We compare the playout schemes of using stream 1 only, using stream 2 only, and using both MD-coded streams. Performance metrics used to evaluate the schemes are the average playout delay and the packet-erasure rate. For the multi-stream scheme, the packet-erasure rate is defined as the percentage of losing both descriptions. The results are shown in Figure 3. The continuous curves with different packet-erasure rate and playout delay are obtained by varying the safety factor β in (2). From it we observe a significant reduction of the packet-erasure rate for a fixed target playout delay when using the multi-stream scheme. At the same average delay of 159 ms, the multi-stream scheme yielded the lowest packet-erasure rate of 1.2%, compared with 5% and 26% for single-stream scheme under light and heavy internet traffic, respectively. On the other hand, if fixing the same packet-erasure rate, the multi-stream scheme also results in the lowest average playout delay.

Our next experiment was conducted to determine whether the improved delay-loss tradeoff could also be realized perceptually. The speech quality resulting from the three playout schemes was evaluated in terms of the PESQ (perceptual evaluation of speech quality). The PESQ algorithm is standardized in ITU-T P.862 [10]. It compares the original and the degraded version of a speech sample to assess the speech quality with a mean opinion score value (MOS), which scales from 1 (bad) to 5 (excellent). The speech samples are spoken by various male and female Mandarin speakers and each sample lasts for about 9 seconds. All speech samples are sampled at 8kHz and digitized in 16-bit PCM format. Each voice packet draws from one encoding frame of 18.75 ms in duration and equals to 338 bytes. For proper reconstruction of continuous speech, individual packets are stretched or compressed using our proposed time-scale modification scheme. The PESQ scores versus delay for the three schemes are shown in Figure 4. Compared with single-stream schemes, the better speech quality resulting from the multi-stream scheme is clearly demonstrated. This indicates that our proposed multi-stream scheme improves the delay-loss tradeoff without compromising the speech reconstruction quality.

6. CONCLUSIONS

This paper studies the combined use of packet path diversity and adaptive playout scheduling for reliable voice communication over IP networks. We first formulate the adaptive playout scheduling of multiple streams as a constrained optimization problem leading to a better tradeoff between the packet loss and the end-to-end delay. The results also suggest a new approach to time-scaling of individual packets without changing their acoustic attributes. Experimental results indicate that the proposed technique can improve the delay-loss tradeoff as well as speech reconstruction quality.

7. ACKNOWLEDGEMENTS

This study was jointly supported by MediaTek Inc. and National Science Council, Republic of China, under contracts NSC 94-2218-E-009-021.

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

[3] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide-area networks," in *Proc. IEEE INFOCOM '94*, June. 1994, vol. 2, pp. 680–688.

[4] 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.

[5] Y. J. Liang, N. Farber, and B. Girod, "Adaptive playout scheduling and loss concealment for voice communication over IP networks," *Journal*, vol. 5, pp. 11–25, Dec. 2003.

[6] A. Shallwani and P. Kabal, "An adaptive playout algorithm with delay spike detection for real-time VoIP," in *Proc. IEEE Canadian Conf. Elec. Comp. Eng.*, Montreal, Canada, May. 2003.

[7] T. F. Quatieri and R. J. McAulay, "Shape invariant time-scale and pitch modification of speech," *IEEE Trans. Signal Process*, 40 (3), pp. 497–510, 1992.

[8] W. Verhelst and M. Roelands, "An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech," in *Proc. ICASSP 93*, April. 1993, vol. II, pp. 554–557.

[9] J. C. Bolot, "Characetrizing end-to-end packet delay and loss in the internet," *Joural of High-Speed Networks*, vol. 2, pp. 305-323, Dec 1993.

[10] ITU-T Recommendation P.862, *Perceptual evaluation of speech quality (PESQ), and objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*. Feb. 2001.

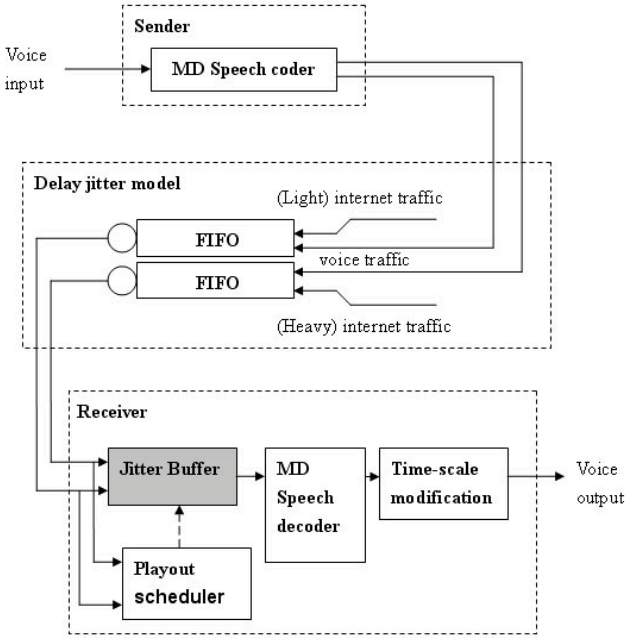

Figure 1: The multi-stream voice communication system

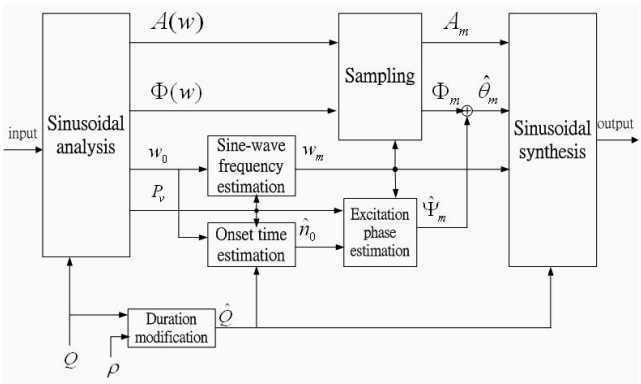

Figure 2: The time-scale modification system

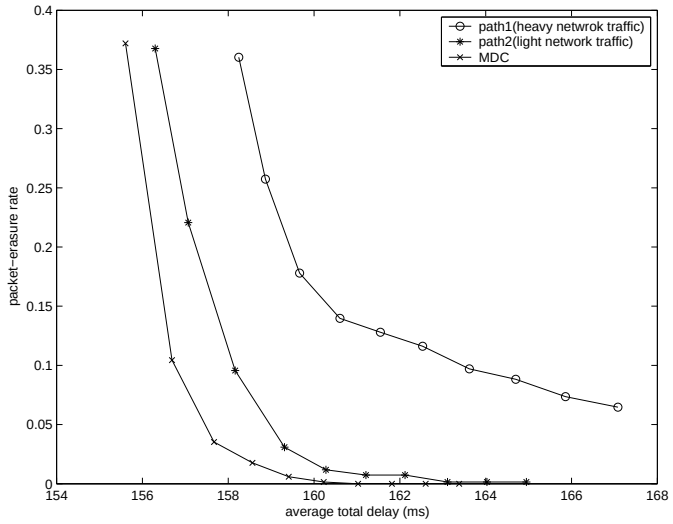

Figure 3: Delay-erasure performance of adaptive playout algorithms

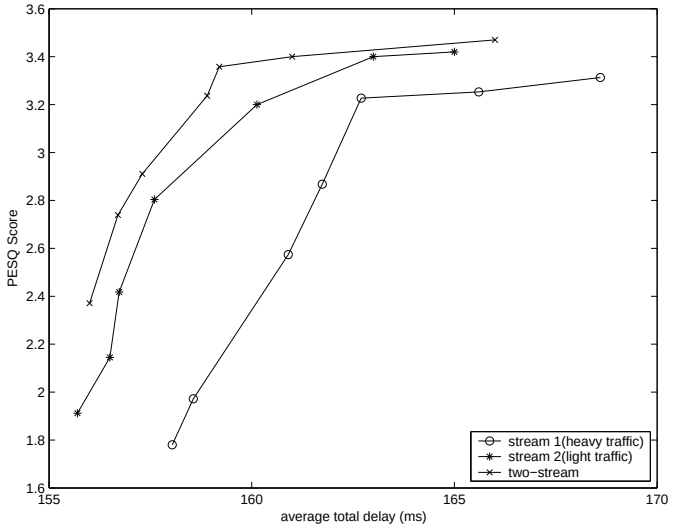

Figure 4: PESQ performance of adaptive playout algorithms