Quality-Based Playout Buffering with FEC for Conversational VoIP

Qipeng Gong, Peter Kabal

Electrical & Computer Engineering, McGill University
Montreal,Quebec,Canada H3A 2A7
qi.gong@mail.mcgill.ca peter.kabal@mcgill.ca

Abstract

In Voice-over-IP, buffer delay and packet loss are two main factors effecting perceived conversational quality. A quality-based algorithm aims to seek an optimum balancing of delay versus loss. To improve perceived quality further, steps should be taken to mitigate the effect of losses due to network (missing packets) and buffer underflow (late packets) without increasing buffer delays. In this paper, we propose a quality-based playout algorithm with an FEC design based on conversational quality including calling quality and interactivity. The simulation results show our algorithm's efficiency of correcting for losses (isolated and burst) and improving perceived conversational quality.

1. Introduction

Voice over IP (VoIP) allows voice to be integrated into a data network. The main challenge facing VoIP is how to provide voice quality equal to that provided by the traditional switched telephone network, i.e. PSTN. However, the IP service model is "best effort", which makes no guarantee on quality. Therefore, for real-time voice communication over IP, the requirements on delay and packet loss are stringent to maintain proper Quality of Service. Efforts have been made in the literature to reduce the delay, smooth the delay variation, and conceal packet losses [1].

A jitter buffer is introduced at the receiver side to compensate for the delay jitter that appears in packet-based networks. The size of this buffer can be fixed or adaptive. In buffering VoIP packets at the receiver, delay is traded against packet losses. However, a long buffer increases the conversational delay which impedes the interactivity of conversations. The interactivity of the conversation is considered to be transparent if the end-to-end delay is less than 150 ms [2]. The ITU-T recommends that the upper limit of end-to-end delay is 400 ms [2]. For applications which experience long network delays, it is desirable to keep the size of jitter buffers small to avoid adding additional delay. Many solutions have been proposed to design the jitter buffer – reference [3] gives a survey and an analysis of several approaches.

Since 2003, several quality-based algorithms have been developed which consider both losses and delays, e.g., [4], [5], [6], [7]. The basic idea is to seek an optimum balancing of the delay versus the loss based on ITU-T E-Model quality measurement. However, these algorithms are still subject to packet losses under certain network conditions. The packet losses includes both network losses (packets that never arrive) and late packets (buffer underflow). If packet losses exhibit burstness, degradation on perceived quality is more than that caused by isolated losses. In Section 2, we show the effect burst length on perceived quality. The task turns into how to reduce the effect of packet losses (isolated and burst losses) without increasing the size of jitter buffers.

Forward error correction (FEC) [8] is used to mitigate the impact of packet losses by sending redundant information. There are two types of FEC schemes: *media-independent FEC* and *media-dependent FEC* (also known as *signal processing FEC* (SP-FEC)). Media-independent FEC uses block codes to provide redundant information, while SP-FEC piggybacks the redundant information onto the subsequent packets. In this paper, we use SP-FEC to avoid the increased delay at the sender imposed by block coding.

To use the redundant information in SP-FEC, the decoder must implement a delay. However, since a jitter buffer is already present at the receiver, there need be no additional delays if SP-FEC is integrated with the jitter protection algorithm.

The redundant information implies an increase in bit rate. To keep the rate down, the redundant information can be encoded more compactly, perhaps entailing a small loss in quality which only comes to play during packet losses. To lower the overall bit rate, separate *primary* and *redundant* encoding can be used to code the redundant information using a lower rate-compression method, resulting in a lower quality for the recovered packet [9]. For example, G.711 (64 kb/s) as the main payload can be combined with GSM (13 kb/s) or G.729 (8 kb/s) for the redundant information. It is to be noted that the speech payload of VoIP is small and so even for G.711, the overhead for a 20 ms IP packet is 25%. For the lower rate coders, the overhead is much larger. Doubling the payload does not double the packet length.

In this paper, we propose a new quality-based playout scheme algorithm with SP-FEC. At a sender's side, m previous voiced packets are added to the packet. The piggybacking stops whenever "hangover" is detected. The value of m is specified by an RTCP packet sent from the receiver. This buffer-aware FEC scheme avoids sending redundant information which cannot be used by the receiver. At the receiver side, a quality-based jitter buffer is used for optimum perceived quality and low conversational delay under the measured network conditions. We use the same buffer design as our previous work in [7]. As a proof of concept and for simplicity, both the primary and redundant information is encoded using G.711. Unlike other FEC schemes which send an RTCP packet at regular intervals (e.g., 5 ms) to adapt the number of piggybacked packets, we send an RTCP packet at the beginning of a talkspurt if the difference of current jitter buffer and previous one is more than one packet length.

The contributions of this paper are three-fold:

1. investigate the effect of burst losses on perceived quality using latest PESQ
2. a new buffer aware SP-FEC scheme

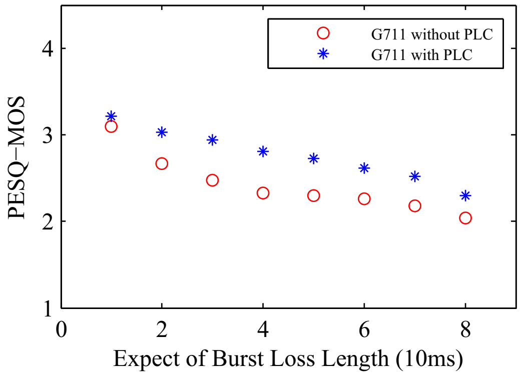

Figure 1: Expected burst length vs. PESQ-MOS.

3. a quality-based playout scheme with this new SP-FEC scheme

The paper is organized as follows: in Section 2, we show the effect of burst losses on perceived quality assessed by PESQ; in Section 3.1, we propose a new buffer aware SP-FEC scheme at the sender's side using buffer information from RTCP packet sent from the receiver's side; a quality-based playout schedule algorithm with buffer aware SP-FEC is proposed in Section 3.3. Finally, simulations and conclusion are presented in Section 4 and Section 5.

2. Effect of Burst Loss Length on Perceived Quality

Packet loss is a main factor influencing perceived quality. Most playout algorithms try to reduce packet loss rate (PLR) to improve perceived quality. Packet loss concealment (PLC) techniques are used to generate lost packets. In this section, we investigate the effect of burst packet loss on perceived quality. Perceived quality is calculated objectively using PESQ [10]. Note that PESQ only measures quality ignoring delays.

We randomly select 20 speech files (10 male, 10 female) from our speech database. Each speech file is 2–3 s in duration. A 2-state Gilbert model is used for the packet loss process. The transition probabilities are set such that the packet loss is 5% and that an expected burst length ($E[BL]$) is achieved. The $E[BL]$ is varied from 1 frame to 8 frames (10 ms for each frame). When $E[BL] = 1\times$frame, the packet loss is random, with no burst loss. For each $E[BL]$, we generate losses for each file and calculate the PESQ-MOS scores using PESQ [10], and then average the scores. Figure 1 shows that PESQ-MOS scores decline with the increment of $E[BL]$. In Figure 1, we also compare the quality between two cases: G.711 PLC algorithm (see G.711 Appendix I for details) for the missing packets and silence substitution of lost packets. It is shown that G.711 PLC algorithm improves perceived quality, but the quality still drops down when $E[BL]$ increases. Therefore, when packets are lost successively in a long burst or when network delays suddenly increase for a period of time, PLC techniques are not entirely effective.

Therefore, burst losses degrade perceived quality even though PLC algorithms are used. To improve quality for VoIP, steps should be taken to reduce burst losses

3. Playout Scheduling Algorithm using SP-FEC

For VoIP applications, the call quality is of the most concern. For conversational VoIP, conversational delay plays an important role on perceived quality. A long conversational delay breaks up the interactivity of a conversation. With conversational interactivity in mind, we have proposed a new quality measurement for conversational VoIP in [7], which takes into account both voice quality and conversational delays. In this paper, we use it as our optimization criterion for the design of playout scheduling. To improve quality further, we use the new SP-FEC to reduce packet losses.

3.1. A New SP-FEC Scheme

SP-FEC works at the send side to send redundant information to enable recovery of missing packets. To safeguard against burst losses, m previous packets are piggybacked onto the current packet.

Since SP-FEC works with jitter buffering at the receiver's side, no additional delay is needed. However, the size of jitter buffer influences the efficiency of recovering lost packets. The reconstruction from redundant information is possible only when the buffer size is greater than the time interval between the lost packet and the packet containing the corresponding redundant packet. In other words, the packet with the lost packet must arrive before the playout time scheduled for the lost packet. For example, if n-th packet is lost, it may be recovered with the piggybacked packet in the $(n+1)$-th packet only if the jitter buffer size is greater than T_p (T_p is the duration of speech segment packetized in one voiced packet), and can be recovered using $(n+2)$-th packet only if jitter buffer size is greater than $2 \times T_p$, etc. Therefore, it is reasonable to vary the number of redundant packets at the send side according to the jitter buffer size at the receiver's side. We only piggyback the voiced packets in talkspurts, and stop piggybacking whenever the "hangover" is detected (VAD/DTX from the G.729 [11]).

Our SP-FEC scheme is

- At the sender's side: at the beginning of a talkspurt, piggyback previous m voiced packets, m is calculated according to the latest RTCP packet which contains the information of jitter buffer at the receiver's side, stop piggybacking whenever the "hangover" packet is detected.
- At the receiver's side: at the beginning of a talkspurt, send an RTCP packet if current jitter buffer is changed greater than one packet length compared to the one for the previous talkspurt.

3.2. E-Model-based Conversational Quality Maximization

In this paper, we use the conversational quality measurement proposed in [7] as

$$Q_c = R + g(D_c). \tag{1}$$

where R is ITU-T E-model R factor, D_c is the conversational delay which is calculated using the method proposed in our previous work [7]. Although $g(\cdot)$ is unknown so far, the relation between Q_c and D_c is: Q_c goes down when D_c goes up, and vice versa. Hence, maximization of Q_c is equal to maximizing the R factor and minimizing D_c.

According to [12], the R factor can be written as

$$R = 93.2 - I_e - I_d, \tag{2}$$

where I_d is the delay impairment factor, and I_e is the equipment impairment factor. I_d can be derived by a simplified fitting process from [4],

$$I_d = 0.024d + 0.11(d - 177.3)\,H(d - 177.3), \qquad (3)$$

where

$$H(x) = \left\{ \begin{array}{ll} 0, & x < 0 \\ 1, & x \geq 0. \end{array} \right.$$

The equipment impairment factor is codec dependent. For G.711 with PLC, it can be approximated as [5]

$$I_e = I_{ec} + I_\rho = 0 + 7\ln(1 + 50\rho), \qquad (4)$$

where I_{ec} is the impairment caused by encoder, which is 0 for G.711, and ρ is the packet loss including network loss and the loss caused by jitter buffer.

According to [7], (2) can be written as

$$\begin{aligned} R &= 93.2 - (I_d + I_\rho) \\ &= 93.2 - \big(0.024d + 0.11(d - 177.3)\,H(d - 177.3) \qquad (5) \\ &\quad + 7\ln\big(1 + 50(\rho_n + \rho_d)\big)\big), \end{aligned}$$

with ρ_n is the network loss and ρ_d is the loss caused by buffer, which depends on the playout delay. ρ_d can be calculated as

$$\begin{aligned} \rho_d &= (1 - \rho_n)P(X > d) = (1 - \rho_n)(1 - P(X \leq d)) \\ &= (1 - \rho_n)(1 - F(d)), \end{aligned} \qquad (6)$$

in which $F(d)$ is the cumulative distribution function (CDF) of delay. In this paper, $F(d)$ is calculated as a function of playout delay using the histogram of the most recent w packet delays. In our simulation, $w = 1000$ packets.

3.3. New Playout Scheduling Algorithm

Human Speech consists of silence and one or more talk-spurts. Packet losses during talk-spurts decreases the perceived quality dramatically, while losses during silence period cause almost no effect on the perceived quality. Therefore, many playout scheduling algorithms tune a jitter buffer at the beginning of each talk-spurt. Compared with continuously updating approaches, a per-talkspurt approach takes the advantage of producing a smoother playout voice.

As in [7], a steady-state buffer depth d_{jitter} is calculated for each talkspurt by maximizing the R factor in (5), and the conversational delay D_c is reduced by two steps. The first packet of a talk-spurt is stretched and played out as soon as it arrives. This stretching process increase the buffer depth. Second, at the end of a talkspurt, compress the voiced packets in a jitter buffer whenever the "hangover" packet is detected.

The following operations are performed:

- During a silence period, comfort noise is played out every 10 ms, no matter whether the SID packet arrives or not. The jitter buffer size is zero. Information about occurred packet losses and transmission delay are stored. SID packets are used to update the comfort noise parameters.
- When the first voiced packet of the first talk-spurt arrives, packet WSOLA (PWSOLA) [13] is applied to stretch the decoded speech before it is played out. The jitter buffer size increases by $(\alpha - 1) \times T_F$ (α is the stretch factor, and T_F is the payload length of a packet). The d_{jitter} parameter is estimated based on previously stored packet delay information (window size is 1000 packets), send an RTCP packet with the information of d_{jitter} if the absolute difference between d_{jitter} and previous d_{jitter} is greater than T_p.
- When the estimated d_{jitter} is achieved, the decoded speech is not stretched any further. The depth of jitter buffer keeps the steady-state value d_{jitter} and $\alpha = 1$.
- At the end of a conversation turn, when the hangover is detected, PWSOLA is applied to compress the decoded speech before it is played out. The jitter buffer size decreases by $(1 - \alpha) \times T_F$. Compression stops when jitter depth is decreased to zero. It is possible for "hangover" to happen in the middle of the talk-spurt, for example, during the silence gap within a word. In this case, we stretch the subsequent voiced packet as if it were the beginning of the talk-spurt. A noticeable change in the silence gap can be avoided [14].

In this paper, for simplicity, we use the same G.711 encoder for original and redundant descriptions. Then a packet is played out either the packet itself or the following packets piggybacked with it are received.

4. Results

To simulate the transmission over the internet, we use three delay trace files, two (Trace 2 & Trace 3) from [4] between UK and China, one (Trace 1) was collected in January, 2009 from McGill University (Canada) to Shanghai JiaoTong University (China) (see [7] for details). For saving space, we use Table 1 to show the main characteristics of the traces used in this paper. The network packet loss is modeled by a 2-state Gilbert Model. The network loss rate is 2% and the expect of network burst loss is 2 packets. The conversation used in our simulation is from the recording of a real dialog (with noisy background), which consists of conversation turns, in an "ask-response" pattern.

Table 1: Network Delay Traces

Trace	Min (ms)	Average (ms)	Max (ms)
Trace1	153	154	221
Trace2	118	145	615
Trace3	122	186	888

The proposed algorithm (VJM_FEC) is compared with other 4 algorithms: Exponential-average (Exp-Avg) [15], Fast exponential average (Fast-Exp) [15], Sun's quality-based algorithm (Sun_opt) [4], the buffering algorithm in our previous work (VJM_adaptive) [7]. The results are shown in Table 2. In all three channels, Fast-Exp gets the highest PESQ-MOS score, but suffers from very a high conversational delay. Exp-Avg obtains lowest conversational delay in Trace 1, but the lowest quality because more late packets are dropped due to buffer underflow. In Trace 2, Exp-Avg achieves better overall quality than the quality-based algorithm – Sun_opt, but poorer performance than VJM_adaptive and VJM_FEC. Exp-Avg gets relative high PESQ-MOS score in Trace 3 at expense of a large jitter buffer delay, so the conversational delay is higher than quality-based algorithms. Therefore, the quality-based techniques always achieve a good balancing between packet losses and delays. In quality-based algorithms, Sun_opt performs similarly in

PESQ-MOS score as VJM_adaptive with longer conversational delays, because two steps (see 3.3 for details) to reduce conversational delays are used in VJM_adaptive besides optimization of E-Model. VJM_FEC obtains higher PESQ-MOS scores than VJM_adaptive with the same conversational delays because SP-FEC is used to reduce packet losses. Overall, the proposed algorithm VJM_FEC performs better than other algorithms and achieves improved conversational quality (high perceived quality and low conversational delay).

Table 2: Performance Comparison of Jitter Buffering Algorithms for Internet Traces

Trace	Buffering algorithms	Conversational delay (ms)	PESQ-MOS	PLR (%)
1	Exp-Avg	321.0	2.60	7.4
	Fast-Exp	510.9	3.08	1.5
	Sun_opt	369.0	2.85	4.0
	VJM_adaptive	343.3	2.90	4.2
	VJM_FEC	343.3	2.93	3.6
2	Exp-Avg	312.8	2.17	16.0
	Fast-Exp	364.4	2.54	8.2
	Sun_opt	326.1	2.14	17.8
	VJM_adaptive	300.1	2.12	18.0
	VJM_FEC	300.1	2.22	15.5
3	Exp-Avg	372.1	2.04	12.0
	Fast-Exp	894.3	2.47	6.0
	Sun_opt	302.6	1.83	15.7
	VJM_adaptive	292.1	1.99	13.9
	VJM_FEC	292.1	2.07	12.5

From the results in Table 2, VJM_FEC performs better than our previous work VJM_adaptive, with the same conversational delay and higher PESQ-MOS score. In Table 3, we calculate the number of burst loss packets for these two algorithms. The results shows the efficiency of VJM_FEC to reduce burst losses.

Table 3: Burst Loss Reduction

	Number of burst loss packets		Reduced burst
Trace	VJM_adaptive	VJM_FEC	Loss(%)
Trace1	23	20	13.04
Trace2	112	97	13.39
Trace3	144	133	7.64

5. Conclusions

In conversational VoIP, customers expect for high-quality service which provides clear, continuous, and interactive conversation. Packet losses and delays (end-to-end delays and conversational delays) are the main factors to influence perceived quality. In this paper, we propose a quality-based playout scheme with SP-FEC. It is based on maximizing calling quality and reducing conversational delay. SP-FEC is used to recovery packet losses and reduce the effect of burst losses on perceived quality. The results of our simulation shows that perceived quality is improved by proposed algorithm.

6. References

[1] Y. J. Liang, N. Farber, and B. Girod, "Adaptive Playout Scheduling using Time-Scale Modification in Packet Voice Communications," in *IEEE ICASSP '01*, Salt Lake City, USA, Jan. 2001, pp. 1445–1448.

[2] ITU-T, *Recomendation G.114: One-way Transmission Time*, ITU Std., May 2003.

[3] L.Atzori and M. L. Lobina, "Playout Buffering in IP Telephony: A Survey Discussing Problems and Approaches," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 3, pp. 36–46, 2006.

[4] L. Sun and E. Ifeachor, "New Models for Perceived Voice Quality Prediction and their Applications in Playout Buffer Optimization for VoIP Networks," in *Proc. IEEE ICC*, Paris, France, Jun. 2004, pp. 1478–1483.

[5] M. Ghanassi and P. Kabal, "Optimizing Voice-over-IP Speech Quality Using Path Diversity," in *Proc. IEEE Workshop on Multimedia Signal Processing*, Victoria,BC,Canada, Oct. 2006, pp. 155–160.

[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*, vol. 11, no. 3, pp. 382–385, Mar. 2004.

[7] Q. Gong and P. Kabal, "A New Optimum Jitter Protection for Conversational IP," in *IEEE International Conference on Wireless Communications & Signal Processing*, Nanjing, China, Nov. 2009, pp. 1–5.

[8] J-C. Bolot, S. Fosse Parisis, and D. Towsley, "Adaptive FEC-based Error Control for Internet Telephony," in *IEEE Infocom'99*, vol. 3, New York, USA, Mar. 1999, pp. 1453 – 1460.

[9] J. Benesty, M. M. Sondhi, and Y. Huang (Eds.), *Springer Handbook of Speech Processing*. Berlin Heidelberg: Springer-Verlag, 2008.

[10] ITU-T, *P.862: Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs*, ITU Std., Nov. 2005.

[11] ITU, *Recommendation G.729: Coding of speech at 8 kbit/s using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)*, ITU Std., Jan. 2007.

[12] ITU-T, *The E-Model, a computational model for use in transmission planning*, ITU Std., Mar. 2003.

[13] Y.-J. Liang, N. Farber, and B. Girod, "Adaptive Playout Scheduling and Loss Concealment for Voice Communication over IP Networks," *IEEE Transactions on Multimedia*, vol. 5, no. 4, pp. 532–559, Dec. 2003.

[14] M. Lee, J. McGowan, and M. C. Recchione, "Enabling Wireless VoIP," *Bell Labs Technical Journal*, vol. 11, pp. 201–215, Nov. 2007.

[15] K. Fujimoto, S. Ata, and Murata, "Adaptive Playout Buffer Algorithm for Enhancing Perceived Quality of Streaming Applications," *Telecommunication Systems*, vol. 25, no. 3-4, pp. 259–271, 2004.