Prediction of Perceived Conversational Speech Quality and Effects of Playout Buffer Algorithms

Lingfen Sun and Emmanuel C. Ifeachor

Department of Communication and Electronic Engineering
University of Plymouth
Plymouth PL4 8AA, U.K.

Abstract*— Perceived conversational speech quality is a key quality of service (QoS) metric for voice over IP (VoIP) applications. Speech quality is mainly affected by network impairments, such as delay, jitter and packet loss. Playout buffer algorithms are used to compensate for jitter, based on a tradeoff between delay and loss, but can have a significant effect on perceived quality. The main aim in this paper is to assess how buffer algorithms affect perceived speech quality and how to choose the best algorithm and its parameters to obtain optimum perceived speech quality (in terms of an objective Mean Opinion Score). The contributions of the paper are three-fold. First, we introduce a new methodology for predicting conversational speech quality (conversational Mean Opinion Score or MOSc) which combines the latest ITU-T speech quality measurement algorithm (PESQ) and the concepts of the E-model. Second, we assess different playout buffer algorithms using the new MOSc metric on Internet trace data. Our findings indicate that, in general, end-to-end delay has a major effect on the selection of a buffer algorithm and its parameters. For small end-to-end delays, an algorithm that seeks to minimise loss is preferred, whereas for large end-to-end delays, an algorithm that aims at a minimum buffer delay is best. Third, we propose a modified buffer algorithm together with an adaptive parameter adjustment scheme. Preliminary results show that this can achieve an "optimum" perceived speech quality for all the traces considered. The results are based on Internet trace data measurements between UK and USA, UK and China, and UK and Germany.

Keywords- Voice over IP; Conversational Speech Quality; Playout Buffer Algorithm; Jitter; Packet Loss; Perceived Quality

I. INTRODUCTION

IP networks are on a steep slope of innovation that will make them the long-term carriers of different types of traffic including speech, but they are not designed to support real-time voice communication. In voice over IP (VoIP) applications, delay, jitter (i.e. delay variation) and packet loss are the main network impairments that affect perceived speech quality. Jitter can be partially compensated for by using a playout buffer at the receiving end, but this introduces further delay (buffer delay) and additional packet loss (packets arriving after their playout times will be dropped by the receiver). A tradeoff is necessary between increased packet loss and buffer delay to achieve a satisfactory result for any playout buffer algorithm. For example, the longer the buffer delay, the lower the late arrival loss and vice versa.

In the past, the choice of a buffer algorithm was purely based on buffer delay and loss performance (e.g. minimum end-to-end delay for a given packet loss rate [1,2,3] or minimum late arrival loss [1]). Given that the ultimate purpose of a buffer algorithm is to obtain a better perceived speech quality, this approach is inappropriate, as it does not provide a direct link to perceived speech quality. From QoS perspective, the choice of the best buffer algorithm for a given situation should be determined by the likely perceived speech quality. This issue is now recognized [8], but the work so far has been limited to one adaptive buffer algorithm and a fixed parameter. It is still unclear how different buffer algorithms and parameters affect perceived quality and how to determine the buffer algorithm/parameters to achieve the optimum perceived speech quality (in terms of an objective MOS).

Perceived speech quality during a VoIP communication can be expressed as a conversational Mean Opinion Score (MOSc). MOSc values may be obtained by subjective listening tests [4] or by objective measurement methods, such as the ITU E-model [5]. The E-model (or the Extended E-model) has been widely used for measuring and assessing conversational speech quality for VoIP applications [6,7,8]. It is based on the principle that the perceptual effects of different impairments are additive on a psychophysical scale. As the E-model consists of very complicated equations and is only applicable to a limited number of codecs at present, we have developed a more general method to predict MOSc. The new method combines the latest ITU-T perceived speech quality measurement algorithm (PESQ) [9] and the concepts of the E-model. The method is suitable for any codec (from 64Kb/s to 4Kb/s) [9] and may also be used to monitor/predict conversation speech quality in practice. The accuracy of the method is limited mainly by the accuracy of the PESQ algorithm, but this can be readily replaced by, for example, the next generation PESQ algorithm, if necessary.

The main contributions of this paper are threefold. First, we introduce a new methodology to predict MOSc score, based on a combination of PESQ and E-model. Second, we assess different buffer algorithms/parameters using the new MOSc instead of existing packet loss/delay metric. Third, we propose a modified buffer algorithm and an adaptive parameter adjustment scheme which can achieve an "optimum" perceived quality for different categories of traces. To assess the quality of current VoIP networks and to evaluate the performance of

The work is sponsored in part by Acterna.

buffer algorithms, we collected Internet trace data between UK and USA, UK and China, and UK and Germany.

The remainder of the paper is structured as follows. Section II presents the method used to collect the Internet trace data, the data, and the conversational speech quality measurement method. Section III compares and analyses the performance of different buffer algorithms and parameters using the new MOSc metric. In Section IV, we present a modified buffer algorithm and an adaptive parameter adjustment scheme. Section V concludes the paper.

II. DATA COLLECTION AND MEASUREMENT

A. Internet trace data collection

We use a UDP/IP probe tool [10] to collect and measure the main network parameters that affect voice quality. It consists of a client/server program, which runs in a local host and at a remote host. It transmits a stream of UDP/IP packets over the network to emulate VoIP traffic, and at the remote host the packets received are echoed back to the local host. Each packet has a sequence number, which indicates the order the packets were sent and can be used to deduce packets that have been lost in the network. The timestamps can be used to deduce network delay and delay variations. Similar tools have been used for experimental assessment of end-to-end behavior of Internet in the past [12,13,14] and more recently for speech quality prediction [8]. In our experiments, the size of the probe packets is set to 32 bytes. The interval between successive packets is 30 ms, which is similar to G.723.1.

In determining one-way delay, the collected trace data is first processed to remove the differences between the clocks at the two hosts and clock drift (or clock skew) [11]. Further, the data is processed to contain talkspurts and silences using a well-known on/off model with an exponential distribution [15]. A mean of 1.5sec for both talkspurts and silences is selected as in [8,16]. For about a week, we collected trace data from Internet connections between the University of Plymouth (UoP), U.K and Columbia University (CU), USA; between UoP and Beijing University of Posts & Telecommunications (BUPT), China and between UoP and Darmstadt University of Technology (DUT), Germany. These sites were selected because they are international connections with different delay characteristics. The basic information for 4 selected traces with a duration of 30 min (1800 sec) is listed in Table 1. Examples of traces (after synchronisation, deskewing and talkspurt/silence processing) are shown in Fig. 1 (a) to (d).

TABLE I. BASIC INFORMATION FOR TRACES #1 TO #4

Trace Num. #	Trace Path	Start Time (Sender)	Average Network Delay (ms)	Average Packet Loss (%)
1	BUPT → UoP	16:50pm, 0706/02, Fri	255	1.8
2	UoP → CU	13:22pm, 13/04/02, Sat	46	0.3
3	UoP → BUPT	9:11am, 11/06/02, Tue	186	14.2
4	UoP → DUT	17:44pm, 10/06/02, Mon	16	4.2

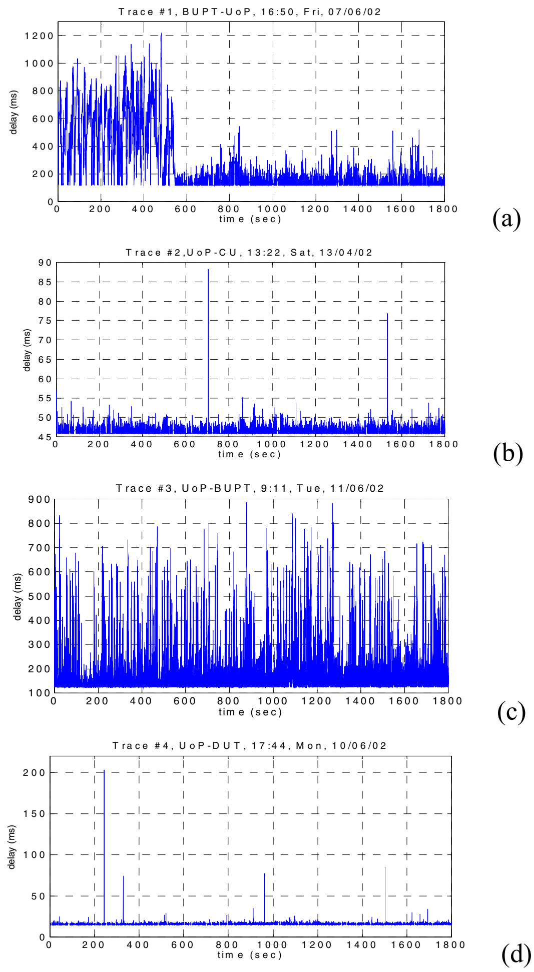

Figure 1. Trace data #1 to #4 ((a) to (d))

We classify traces into two major categories. The first category are traces with small end-to-end delay/jitter, such as traces from UoP to CU and UoP to DUP (Fig 1 (b) and (d)). The other are traces with large/medium delay/jitter, such as those between UK and China (Fig 1 (a) and (c)).

B. Conversational Speech Quality (MOSc) Measurement

The methodology for conversational speech quality measurement is based on the PESQ and the E-model (see Fig. 2). The reference speech signal is first encoded using a suitable codec (e.g. G.723.1) and then processed in accordance with the loss characteristics of the trace data to generate the degraded speech (equivalent to IP impaired speech). In practice, the relevant parameters (i.e. end-to-end delay, delay variation and packet losses) can be obtained from analysis of the RTP header and RTCP report. The ITU voice test signals [18] are used as the reference speech data in our study. The reference speech and degraded speech are then fed to the PESQ to obtain a measure of speech quality due to loss and codec. PESQ is designed for one-way listening-only perceived quality measurement and does not consider the effects of delay, which is required for conversational speech quality. The E-model concepts are used to combine the effects of loss and delay to obtain an overall quality score, MOSc.

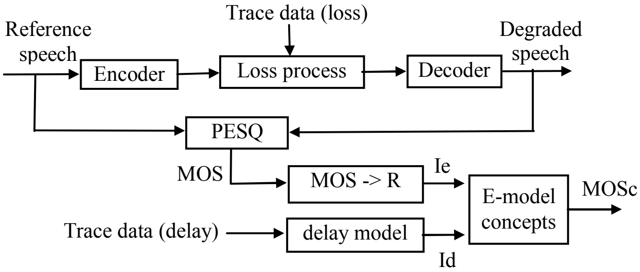

Figure 2. Schematic diagram for MOSc measurement

The PESQ is an intrusive, end-to-end measurement algorithm and requires a reference signal. However, provided a suitable local reference is available, it can be used for non-intrusive measurement [17] to exploit its greater accuracy and applicability to a wide range of codecs. We have followed this approach and extended it to account for delay in the study of the effects of buffer algorithms on IP speech quality.

Ignoring the effects of other impairments (e.g. echo), the rating scale for the E-model, R, may be simplified as follows:

$$R = R_0 - I_d - I_e \qquad (1)$$

where R_0 is the optimum quality value (the default value for R_0 is 93.2 [5] which is used in the study). I_e is known as the equipment impairment factor and accounts for impairments due to non-linear codec and packet loss. I_d accounts for echo and delay. Under perfect echo cancellation conditions, I_d can be calculated by (2) [5].

$$I_d = 0 \quad \text{for} \quad T_a < 100ms \qquad (2a)$$

$$I_d = 25\left\{(1+x^6)^{1/6} - 3\left[1+(x/3)^6\right]^{1/6} + 2\right\} \text{for } \mathrm{T_a} \geq 100\text{ms} \qquad (2b)$$

where $x = (\lg[Ta/100])/\lg 2$ and Ta represents absolute delay (or end-to-end delay).

Given the R value, the corresponding MOS score can be obtained using the following relationship: [5].

$$MOS = 1 \quad \text{for} \quad R \leq 0 \qquad (3a)$$

$$MOS = 1 + 0.035R + R(R-60)(100-R)7\times10^{-6} \quad \text{for } 0 < R < 100 \qquad (3b)$$

$$MOS = 4.5 \quad \text{for} \quad R \geq 100 \qquad (3c)$$

Using a similar 3rd order polynomial the expression for transforming MOS to R is given by (4).

$$R = 3.026x^3 - 25.314x^2 + 87.060x - 57.336 \qquad (4)$$

where x represents MOS value.

Values of MOS obtained from PESQ are first transformed to R using (4) and then to I_e ($I_e = R_0 - R$). The overall score, MOSc, is obtained from (1) and (3).

For every 9 sec trace data (9 sec is chosen because it is within the recommended length for PESQ algorithm [9]), the overall packet loss (including late arrival loss) and overall end-to-end delay (including buffer delay) are calculated based on the playout buffer algorithm used (see next section for details of buffer algorithms). An average end-to-end delay (i.e. real delay) for the 9 sec trace data is also calculated and sent to delay model to get delay impairment I_d. PESQ MOS score is also transformed to I_e value. From I_e and I_d values, the conversational speech quality (MOSc) is calculated. The average MOSc score at the end of the selected trace data (30min) is calculated as the overall MOSc score (recency effect was not considered).

III. BUFFER ALGORITHMS AND PERFORMANCE ANALYSIS

Playout buffer can be fixed or adaptive. A fixed buffer cannot adapt to changing network delay conditions and this may result in poor speech quality. Thus, we have focused on adaptive buffer algorithms and adjust the buffer at the beginning of each talkspurt [1][2][3].

The notations used to describe buffer algorithms are defined in Fig. 3. For packet i, we define t_i as the send time; a_i and p_i as the arriving and playout times, respectively. n_i represents network delay and d_i is the actual end-to-end delay or "playout delay". b_i is the buffer delay.

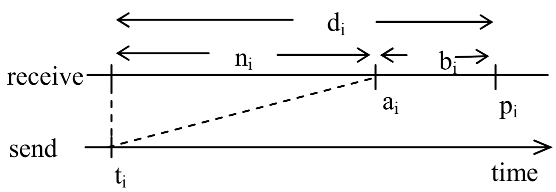

Figure 3. Timing associated with packet i

We first implemented four algorithms proposed by Ramjee et al. [1]. These four algorithms maintain running estimates of the mean and variation of network delay, i.e. $\hat{d}_i$ and $\hat{v}_i$, seen up to the arrival of the ith packet. If packet i is the first packet of a talkspurt, its playout time p_i is computed as:

$$p_i = t_i + \hat{d}_i + \mu * \hat{v}_i \qquad (5)$$

where μ is a constant and $\hat{v}_i$ is given by:

$$\hat{v}_i = \alpha\hat{v}_{i-1} + (1-\alpha)\left|\hat{d}_i - n_i\right| \qquad (6)$$

n_i is the network delay of the ith packet.

The playout delay for subsequent packets (e.g. packet j) in a talkspurt is kept the same as $d_j = d_i$.

The four algorithms differ only in the computation of $\hat{d}_i$.

1) Algorithm 1 ("exp-avg"): This algorithm estimates the mean delay through an exponentially weighted average.

$$\hat{d}_i = \alpha\hat{d}_{i-1} + (1-\alpha)n_i \quad (\text{with } \alpha = 0.998002) \qquad (7)$$

2) Algorithm 2 ("fast-exp"):This algorithm is similar to the first, except it adapts more quickly to increases in delays by using a smaller weighting factor as delays increase:

$$\hat{d}_i = \begin{cases} \beta\hat{d}_{i-1} + (1-\beta)n_i & n_i > \hat{d}_{i-1} \\ \alpha\hat{d}_{i-1} + (1-\alpha)n_i & n_i \le \hat{d}_{i-1} \end{cases} \quad (8)$$

with $\beta = 0.75$ and $\alpha = 0.998002$ as before.

3) Algorithm 3 ("min –delay"): This is more aggressive in minimizing delays. It uses the minimum delay of all packets received in the current talkspurt. Let S_i be this set of delays.

$$\hat{d}_i = \min_{j \in s_i} \{n_j\} \quad (9)$$

4) Algorithm 4 ("spk-delay")

This algorithm contains a spike detection algorithm. During a spike, the delay estimate tracks the delays closely, but after a spike, it is similar to Algorithm 1 (with $\alpha = 0.875$ under Normal mode). We avoid a detailed description here and refer the reader to [1] for details.

There are other more complicated algorithms which can achieve better spike detection than Algorithm 4, such as those mentioned in [2]. As our purpose here is not to find a better algorithm for spike detection, those algorithms are not covered.

In the first experiment, we investigated how the buffer algorithm parameters affect speech quality using MOSc metric. We assume no limitations in buffer size and adapt μ in (5) from 1 to 20, as in [2]. In comparing with the existing performance metrics, we also include the performance of average playout delay (or real delay) and average loss rate (or real loss).

The real delay and loss vs. μ for traces #1 and #2 are shown in Fig 4 (a) to (d), respectively. It is clear that the "fast-exp" has the lowest loss rate but the highest delay for both traces, as it adapts more quickly to increase in delay. The "min-delay" has the lower delay and higher loss for both traces, as it targets at minimum delay. The results for other two algorithms are between that of the "fast-exp" and the "min-delay".

Four buffer algorithms show similar trends at real delay and loss metrics for traces #1 and #2 (similar results obtained for traces #3 and #4). However, the combined effect on perceived quality shows a big difference for two categories of traces (see Figures 4 (e) to (h)). There is an obvious similarity within the same category of traces (e.g. trace #1 and #3, trace #2 and #4). This suggests that the perceived performance of the four buffer algorithms for different parameters is mainly affected by the end-to-end delay/jitter of the trace data.

For small delay/jitter traces, the MOSc score can achieve its "optimum" value when μ is set within a proper range for a certain algorithm (e.g. any μ within 1 to 20 for the "fast-exp" algorithm, and μ > 10 for other three algorithms). The reason behind this is that the end-to-end delay for these two traces does not affect MOSc score, as the overall end-to-end delay is near or less than 100 ms, with the I_d in (2) near to zero. In this case, MOSc is only affected by packet loss and codec.

The performance of the four algorithms differs slightly for traces #1 and #3 (see Fig 4 (e) and (g)). It seems that "min-delay" algorithm can reach the maximum MOSc value for both traces #1 and #3 at different μ values (e.g. μ=6 for trace #1 and μ=2 for trace #3). This maximum MOSc score represents the best overall tradeoff between delay and loss for the selected traces. As the two traces have both large end-to-end delay (over 100ms), delay has a major effect on the perceived speech quality. The "min-delay" algorithm can achieve its good performance as it induces lower buffer delay among the four algorithms. For the "exp-avg" and the "spk-delay" algorithms, there also exists a μ value to achieve a maximum MOSc score, although this maximum value is lower than that of the "min-delay" algorithm. For the "fast-exp" algorithm, MOSc scores just decrease monotonously with μ increasing. This suggests that the impact on speech quality due to buffer delay induced by this algorithm is much higher than the benefits due to lower late arrival loss.

The curves of MOS (from PESQ) and MOSc vs. time for traces #1 and #2 are shown in Fig 5 (a) and (b). Fig 5 (a) is for the "min-delay" algorithm with μ of 6, while Fig 5 (b) is for the "fast-exp" with μ of 2 (both under the best MOSc scenarios). It is almost the same for MOS and MOSc for trace #2, as there is no direct impact from delay, whereas, MOSc is obviously lower than MOS for trace #1 due to the impact from delay.

IV. A MODIFIED PLAYOUT BUFFER ALGORITHM – PERCEIVED QUALITY OPTIMIZATION

From the performance analysis on these two categories of traces, we find that there is no 'best' algorithm/parameter, which can always achieve the 'best' MOSc value for all the traces. However, there is a best algorithm among the four, which is most suitable for each category of traces. For example, the "fast-exp" algorithm is preferred for low delay path/trace within a wide range of μ value (μ within 1 to 20), whereas, the "min-delay" algorithm seems the best for a longer delay trace/path under a certain μ value (μ=6 for trace #1 and μ=2 for trace #3). It suggests that a different algorithm or μ value should be chosen for different traces to achieve an "optimum" perceived quality. Based on this, we propose a modified buffer algorithm which can adapt to the preferred algorithm (e.g. "fast-exp" or "min-delay") automatically according to the running estimate of mean network delay $\hat{d}_i$. The algorithm (abbreviated as "adaptive") is as follows:

if $(\hat{d}_i \ge delay\_threshold)$

$$\hat{d}_i = \min_{j \in s_i} \{n_j\}$$

else { if $(n_i > \hat{d}_{i-1})$

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

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

}

Considering the impact of delay on MOSc (imperceptible when delay is under 150ms [6]), we first set the *delay_threshold* (mean delay) to 150ms and calculate the MOSc score under different μ values (as before). The "adaptive" algorithm can adapt to the "fast-exp" for traces #2 and #4 and to the "min-delay" for traces #1 and #3 (in most cases). The result is the same as that of their adapted algorithms in Fig 4 (e) to (h)).

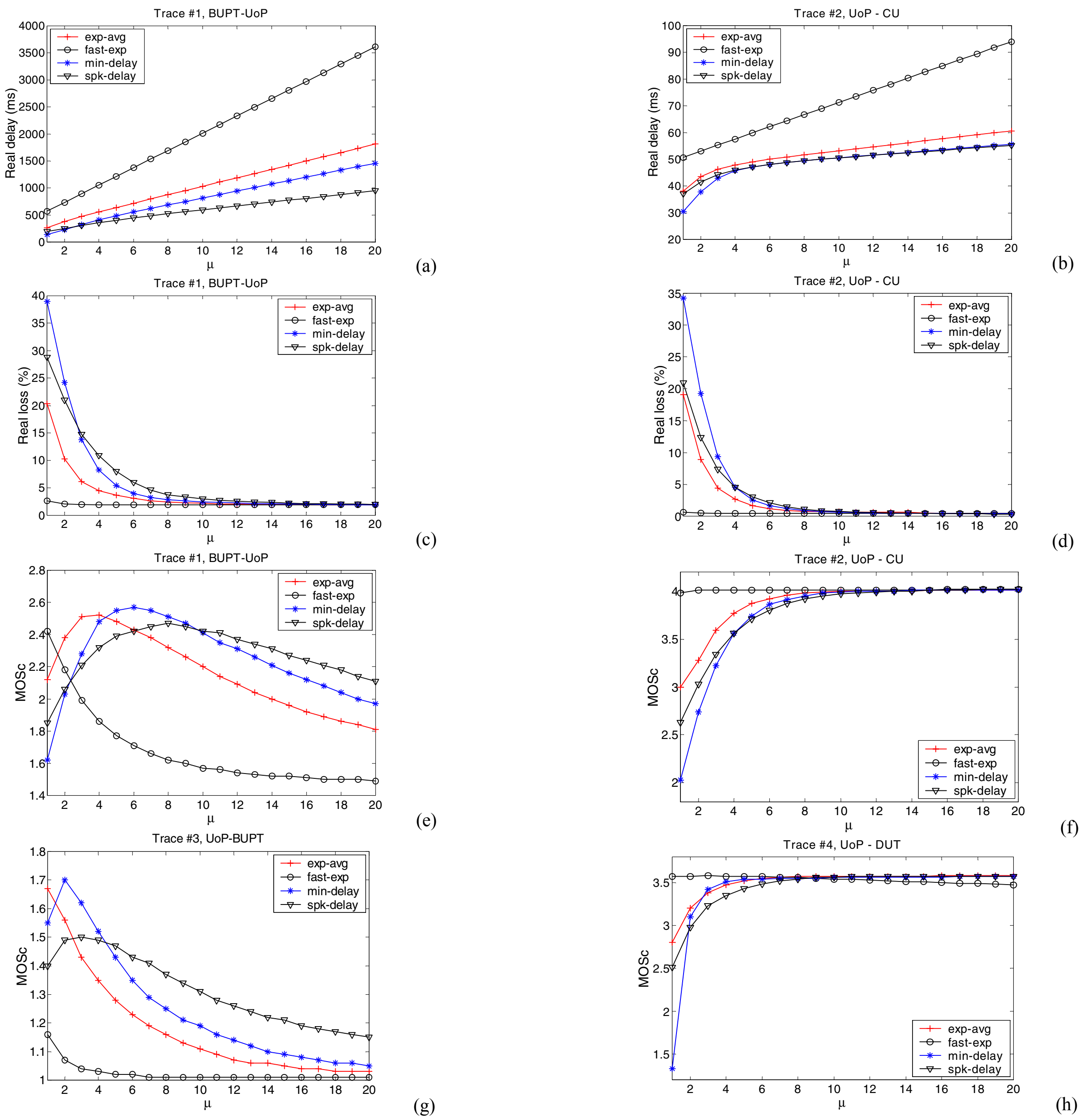

Figure 4. Performance comparison of playout buffer algorithms for Traces #1 to #4

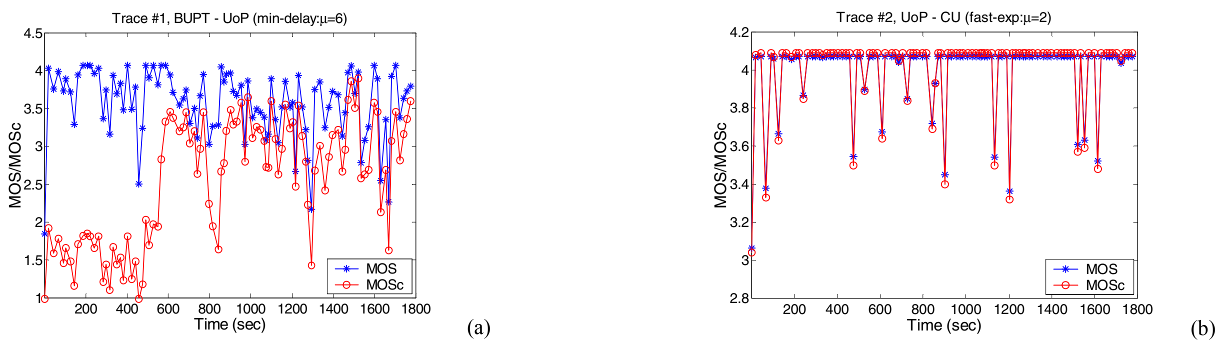

Figure 5. MOS (PESQ) and MOSc vs. time for traces #1 and #2

In order to see how delay threshold affects MOSc, we also set *delay_threshold* to 170, 190, 210 and 250ms and calculate the MOSc score for trace #3 (its average network delay is 186ms as in Table 1). The "adaptive" algorithm swaps between the "fast-exp" and the "min-delay" algorithms according to the change of end-to-end delay. The results for the "adaptive", the "min-delay" and the "fast-exp" algorithms are shown in Fig 6. When the threshold is 170ms, the result of "adaptive" algorithm is similar to that of the "min-delay". With the increase of threshold, the results move towards the direction of the "fast-exp" algorithm and the maximum MOSc score becomes lower.

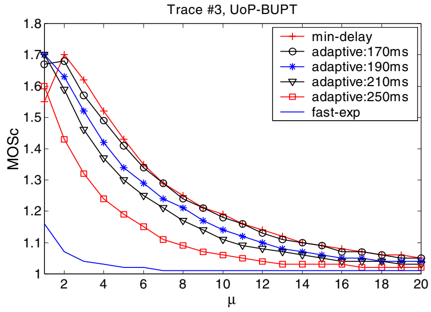

Figure 6. Performance comparison for Trace #3

The results suggest that the "adaptive" algorithm can adapt to the best algorithm for four traces to achieve the best perceived speech quality under the selected delay threshold.

We further investigated how to choose and adapt the parameters (e.g. μ value) of the buffer algorithm to achieve the "optimum" perceived quality all the time. The stages in the adaptation strategy are as follows:

(1). The best μ value (corresponding to the maximum MOSc score) is searched for each test segment (e.g. 9 sec), and this best μ value is used in next segment for the calculation of playout time (p_i) in (5).

(2). For each segment, also calculate MOS_{max}, the maximum PESQ MOS score with only network packet loss.

(3). Search μ ($\mu_{i+1} = \mu_i + 1$, $\mu_{i-1} = \mu_i - 1$), until

$$(((MOSc_{u_i} \geq MOSc_{u_{i-1}}) \wedge (MOSc_{u_i} \geq MOSc_{u_{i+1}})) \vee$$

$(MOSc_{u_i} = MOS_{\max}))$, then, μ_i is the best one for the segment. For the first segment, the search starts from μ =1, for other segment, the search starts from the best μ of the previous segment. If $(MOSc_{u_i} = MOS_{\max})$, the lowest μ that meets these criteria is selected, as this suggests an "optimum" MOSc score with the lowest end-to-end delay.

We implemented this parameter adjustment scheme on the four traces. The preliminary results show that the MOSc score increased obviously for traces #1 and #3. For traces #2 and #4, the MOSc scores always remained at MOS_{max}.

V. CONCLUSIONS

In this paper, we have proposed a new methodology for predicting conversation speech quality (MOSc). We investigated the performance of different buffer algorithms and parameters using the new MOSc metric based on newly collected Internet trace data. Results show that end-to-end delay, in general, has a major effect on the selection of buffer algorithms/parameters. For large to medium end-to-end delay, a buffer algorithm that aims for a minimum delay is preferred, whereas, for small end-to-end delay, an algorithm that targets minimum loss is best. Based on this, we proposed a modified buffer algorithm and an adaptive parameter adjustment scheme. Results show that it can achieve an "optimum" perceived quality for all the traces.

Future work will focus on the analysis of the impact of other parameters (e.g. buffer size) and the impact of parameter adjustment rate on perceived speech quality.

ACKNOWLEDGMENT

We would like to thank Mr. Wenyu Jiang from Columbia University, Mr. Michael Zink from Darmstadt University of Technology, Prof Wendong Wang and Mr. Lunyong Zhang from Beijing University of Posts & Telecommunications for their cooperation in trace data collection.

REFERENCES

[1] R. Ramachandran, J. Kurose, D. Towsley and H. Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide-area networks, Proc. of IEEE Infocom, 1994, vol.2, pp.680 - 688.

[2] S. B. Moon, J. Kurose, D. Towsley, Packet audio playout delay adjustment: performance bounds and algorithms, Multimedia Systems, 1998, vol.6, pp.17 – 28.

[3] J. Rosenberg, L.Qiu and H. Schulzrinne, Integrating Packet FEC into Adaptive Voice Playout Buffer Algorithms on the Internet, Proc. of IEEE Infocom 2000, vol.3, pp.1705 – 1714.

[4] ITU-T P.800, Methods for subjective determination of transmission quality.

[5] ITU-T Recommendation G.107 (05/2000), The E-model, a computational model for use in transmission planning.

[6] TIA/EIA Telecommunications Systems Bulletin, Voice Quality Recommendations for IP Telephony, TSB116, March 2001.

[7] A. Clark, Modeling the Effects of Burst Packet Loss and Recency on Subjective Voice Quality, 2nd IPTel Workshop, 2001, pp.123 – 127.

[8] A. P. Markopoulou, F. A. Tobagi, M.J. Karam, Assessment of VoIP Quality over Internet Backbones, Proc. of IEEE Infocom, 2002.

[9] ITU-T Recommendation 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.

[10] http://www.cs.columbia.edu/~wenyu/

[11] W. Jiang and H. Schulzrinne, QoS Measurement of Internet Real-Time Multimedia Services, Technical Report, CUCS-015-99, Columbia University, Dec. 1999.

[12] D.Sanghi, A.K. Agrawala, O. Gudmundsson, Experimental Assessment of End-to-end Behavior on Internet, Proc. of IEEE Infocom, 1993.

[13] J. –C. Bolot, Characterizing end-to-end packet delay and loss in the Internet, Jour. of High Speech Networks, vol.2, pp. 305-323, 1993.

[14] M. S. Borella, Measurement and Interpretation of Internet Packet Loss, Journal of Communications & Networking, 2(2), June 2000.

[15] P. Brady, A Technique for Investigating on/off Patterns of Speech, Bell Labs Tech. Journal, 44(1):1-22, January 1965.

[16] W. Jiang, H.Schulzrinne, Analysis of on-off patterns in VoIP and their effect on voice traffic aggregation, Proc. of ICCCN 2000.

[17] A. E. Conway, A Passive Method for Monitoring Voice-over-IP Call Quality with ITU-T Objective Speech Quality Measurement Methods, Proc. of IEEE ICC, 2002.

[18] ITU-T Recommendation P.50, Appendix 1, Test signals, 1999.