E-Model based Adaptive Jitter Buffer with Time-Scaling Embedded in AMR decoder

Liyun Pang
Audio Technology Team
Huawei European Research Center
Munich, Germany
liyun.pang@huawei.com

Laszlo Böszörmenyi
Department of Information Technology
University Klagenfurt
Klagenfurt, Austria
office-lb@itec.uni-klu.ac.at

***Abstract*—The premier factors affecting voice quality in packet networks are latency, jitter and packet loss. Jitter buffers are commonly used to counter jitter introduced by queuing in packet switched networks so that a continuous play-out of voice transmitted over the network can be ensured. In this paper, a new adaptive jitter buffer algorithm is proposed. The algorithm consists of an adaptive play-out algorithm based on the extended E-Model with spike detection and a time scaling technique relying on a speech classification mechanism embedded in the AMR decoder. Simulation results show that the proposed algorithm outperforms the best existing algorithms for random packet loss under various network scenarios.**

Keywords - adaptive jitter buffer; E-Model; AMR; time-scaling

I. INTRODUCTION

In recent years, the transport of Voice over IP (VoIP) has gained popularity and is becoming widely used. VoIP can be seen as a replacement to traditional circuit switched telephony with the advantages of cost reduction, simplified network and simplified network management. Voice quality is a critical parameter in the acceptance of VoIP services. Despite the amount of research and development work done in this area, it is still hard to guarantee the same Quality of Service (QoS) as that of traditional telephony. Among the various parameters affecting voice quality, packet losses, latency and delay jitter are the key factors, inevitable in a packet network, contributing to speech quality degradation. In order to balance jitter introduced by queuing in packet switched networks, a jitter buffer mechanism is required at the receiver for ensuring continuous play-out.

When a jitter buffer is applied, received packets are buffered for a while after arrival, and played out sequentially at scheduled time. If some packets arrive after their scheduled play-out time, they are discarded. A late play-out time reduces such kind of packet loss, but introduces unnecessarily long end-to-end delay. The problem of delay jitter is thereby converted into end-to-end delay and packet loss. Previous work mainly focused on designing jitter buffer based solely on the trade-off between end-to-end delay (play-out delay) and packet loss rate due to late arrival. The play-out delay is adjusted either at the beginning of a talk-spurt [1][2] (called per-talk-spurt), or within the speech talk-spurt using time-scale modification to ensure continuous play-out of voice data [3][4] (called per-packet). Although such designs can achieve a minimum average end-to-end delay for a specified packet loss rate, they do not take into account the overall perceived speech quality. Recently, some quality-based approaches have been proposed. Instead of achieving a compromise between delay and packet loss, these approaches adjust the jitter buffer with the objective of optimizing the perceived speech quality given by the Mean Opinion Score (MOS) [5][6]. To develop such quality based approaches, the ITU-T E-Model [13] is one of the most well-known methods. The output of the E-Model, the so called R value, can be easily mapped onto a corresponding MOS value using a transformation given in Appendix I of [13].

Although the ITU-T E-Model has been initially developed for network planning purposes, there has been proposals to extend it not only for evaluating speech quality in conversational communication, but also for monitoring VoIP performance during transmission. In [7][8][9] quality-based play-out scheduling approaches were proposed to maximize perceived speech quality using the R value of the E-Model as cost function. These approaches rely only on adjusting the play-out delay on per-talk-spurt basis. When talk-spurts are long and the network delay varies significantly within them, the performance of these so called talk-spurt-based methods is limited. In [10], a per-packet quality-based jitter buffer algorithm is described. The play-out delay estimation is based on maximizing the R value (or equivalently maximizing the MOS value) and is designed as an unconstrained optimization problem. However, since time-scale modification is required in all per-packet jitter buffer algorithms and a speech frame normally can only be time-scaled within a certain range to avoid degrading voice quality [3], a constrained optimization problem is more suitable.

When designing a quality based algorithm for jitter buffer management, an estimate of network delay distribution is required. Some works assume a certain parametric model to estimate the Cumulative Distribution Function (CDF) of the network delay distribution. For instance, Pareto in [10], Weibull in [7] and Gamma in [11] were used. The use of a certain type of distribution to model

 ISBN: 978-1-61208-127-4

delay behavior in a network is arguable. In fact, delay and jitter in a VoIP session are non stationary and have a high degree of variability even within a single session.

In this paper, a new adaptive jitter buffer system is proposed, implementing per-packet scheduling based on the extended E-Model. In addition, the proposed system contains a spike detection mechanism and a classifier based time scaling technique similar to that proposed in [4]. The time scaling technique is implemented directly inside the speech decoder (AMR decoder [16]) which is advantageous for the quality and makes it possible to use the internal parameters of the codec. For instance, pitch values and gains are particularly useful parameters for time scaling.

The paper is organized as follows. In Section Ⅱ, the extended E-Models used in the proposed jitter buffer system are introduced. In Section Ⅲ, the E-model based play-out algorithm is proposed. Section Ⅳ presents the modified time-scaling embedded AMR decoder. Finally, simulation results illustrating the performance of the proposal and conclusion are presented in Section Ⅴ and Section Ⅵ.

II. Extended E-Model

The ITU-T E-Model is a computational model for the prediction of the expected voice quality which combines different impairments due to codec, echo and other transmission parameters. The underlying assumption of the E-Model is that all impairment factors contributing to speech quality degradation are additive on a psychological scale, and summed to form a rating factor R. The rating factor lies in the range of 0 to 100. A rating of '0' represents a MOS value '1' (bad quality) and '100' of R represents MOS value '4.5' (high quality). The output R value is obtained by subtracting impairment factors from a basic quality measure [13]:

$$R = R_0 - I_s - I_d - I_{e,eff} + A \tag{1}$$

where R_0 represents the basic signal-to-noise ratio; I_s is the Simultaneous Impairment Factor; I_d represents the Delay Impairment Factor; $I_{e,eff}$ is the Effective Equipment Impairment Factor. A is an advantage Factor which has accordingly no relationship with all other parameters and normally can be neglected. All input parameters and their recommended ranges can be found in [13]. For those parameters which are not available at the time of planning, the default values from the ITU [14][15] are recommended. If we only focus on an IP network, the expression of E-Model in (1) can be simplified to the transport layer [12]

$$R = 93.2 - I_d - I_{e,eff} \tag{2}$$

with I_d referring to impairments only due to end-to-end delay d. I_d can be derived by curve fitting as described in [12]

$$I_d = 0.024 \cdot d + 0.11 \cdot (d - 177.3) \cdot H(d - 177.3) \tag{3}$$

where $H(x)$ is the step function ($H(x) = 0$ if $x < 0$; $H(x) = 1$ else). In this paper, we consider only random packet losses for the AMR codec, therefore, $I_{e,eff}$ is obtained either by applying provisioning values from [15]

$$I_{e,eff} = 5 + 90 \cdot (p_n + p_b)/(p_n + p_b + 10) \tag{4}$$

or from the empirical formula [7]

$$I_{e,eff} = 14.96 + 16.68 \cdot ln\big(1 + 30.11 \cdot (p_n + p_b)\big) \tag{5}$$

where p_n is the packet loss rate in the network and p_b is the late packet loss rate dropped by the jitter buffer. Since packets are discarded when they arrive after their scheduled play-out time, the late loss rate p_b is calculated as

$$p_b = (1 - p_n)\big(1 - P_r(X \le d)\big) = (1 - p_n)(1 - F(d)) \tag{6}$$

with $F(d)$ being the CDF of network delay which is obtained in this paper from histogram statistics of previous network delay. If we define the sum of I_d and $I_{e,eff}$ as a new impairment factor I

$$I = I_d + I_{e,eff} \tag{7}$$

then (2) is simplified as

$$R = 93.2 - I \tag{8}$$

This formulation of R (8) is used as the cost function in our jitter buffer management to estimate the play-out delay by maximizing R which is equivalent to minimizing I. Equations (4) and (5) for modeling $I_{e,eff}$ are used both and their performance is compared.

III. Proposed Play-out Algorithm

The proposed receiver includes an adaptive jitter buffer algorithm and the time-scaling embedded in the decoder, as shown in Fig. 1. The Adaptive play-out algorithm is the main control unit. A spike, i.e., the sudden and very high increase of network delay, is very common in VoIP transmission. For this reason, spike detection in [2] is used to switch between NORMAL mode and SPIKE mode. In SPIKE mode, the scheduled play-out time follows current network condition. In NORMAL mode, the scheduled play-out time is based on the delay estimation implementing the extended E-Model, mentioned in Section Ⅱ.

The proposed play-out algorithm will be described using the basic notations listed in Table I.

When a packet arrives at the receiver before its scheduled time, it can be played out without packet loss. Before playing out the current speech frame, the play-out delay of the next coming packet has to be estimated to obtain the modified current frame length. The play-out delay is chosen to maximize the perceived speech quality in terms of R. As discussed in Section Ⅱ, R depends on the end-to-end delay d, network loss rate p_n and buffer loss rate p_b. The buffer loss rate is determined by the play-out buffering algorithm, thus by the end-to-end delay (play-out delay). Therefore (8) can

 ISBN: 978-1-61208-127-4

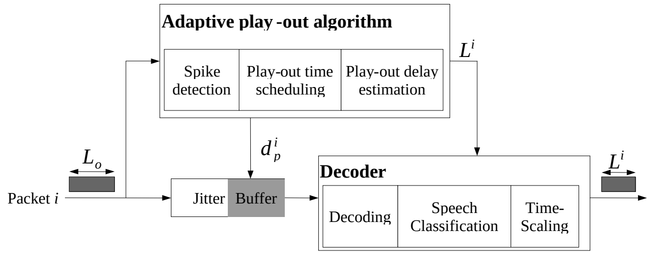

Figure 1. proposed adaptive jitter buffer at the receiver

TABLE I. BASIC NOTATIONS

symbol	Definition
d_n^i	network delay of packet i
d_p^i	actual play-out delay of packet i
$\hat{d}_p^i$	estimated play-out delay of packet i
L_o	original frame length, 160 for AMR
L^i	modified frame length of packet i
Δ^i	frame length difference of packet i
L_{max}	possible maximum time-scaled frame length
L_{min}	possible minimum time-scaled frame length

be expressed as a function of play-out delay, and applied as the cost function in the play-out buffering algorithm to predict the voice quality.

The play-out delay for each packet is estimated based on maximizing the expected R value. The operation of the jitter buffer is based on the statistics of the delay and packet loss of the previous received packets.

The algorithm works as follows

1. Receive a new $packet^i$, obtain network delay information d_n^i from the RTP header information.
2. Spike Detection: check the current network condition, and switch between SPIKE mode and NORMAL mode.
3. Play-out time Scheduling
   a) If this is the first packet of the talk-spurt, follow network delay
   $d_p^i = d_n^i$
   b) Otherwise, use the estimated play-out delay
   $d_p^i = \hat{d}_p^i$
4. Play-out delay Estimation
   a) SPIKE: follow the current network delay $\hat{d}_p^{i+1} = d_n^i$, and skip step 5.
   b) NORMAL: estimate play-out delay based on the E-Model.
5. E-Model based play-out delay estimation in NORMAL mode
   a) Update delay statistics of the most recent received W (history window size) packets only in NORMAL mode
   b) Find the optimal play-out delay for $packet^{i+1}$
   $$\hat{d}_p^{i+1}: I_m\left(\hat{d}_p^{i+1}\right) = \min_{d_{min} \le d \le d_{max}} I_m(d)$$
   where d_{min} and d_{max} are the constraints specified by the time-scaling to make the artifacts less audible:
   $$d_{min} = d_p^i - (L_o - L_{min})$$
   $$d_{max} = d_p^i + (L_{max} - L_o)$$
6. Calculate the new length of packet[i]
   $$\Delta^i = \hat{d}_p^{i+1} - d_p^i$$
   $$L^i = L_o + \Delta^i$$
7. Send $packet^i$ and expected length L^i to the decoder.

IV. TIME-SCALING EMBEDDED IN THE DECODER

The E-Model based play-out scheduling algorithm described in section III is applied specifically to AMR codec. The standard 3GPP AMR decoder is modified to embed a time scaling technique based on speech classification. According to the evaluated frame type, different time-scaling (extension or suppression) operations are applied to the excitation frame which is segmented to four sub-frames.

A. Speech classification

Speech is categorized into silence and talk-spurt, which is further subdivided into voiced/unvoiced. Special frames such as plosive or over-voiced frames are also differentiated from others by using the internal parameters inside the AMR decoder

- Silence/Talk-spurts: The classification between silence and talk-spurts is realized by Voice activity detection (VAD). When operated in DTX mode, Silence and talk-spurts are distinguished from each other by checking if the frame type is SID frame.
- Voiced/Unvoiced: Considering the speech classification implemented in the VMR codec [17], our voiced/unvoiced decision is based on three parameters: The Voicing Factor F_v, Spectral Ratio e_{tilt} and Energy Variation V_e.
  F_v is calculated as an averaged normalized correlation over four sub-frames of speech with the pitch lags T_o. e_{tilt} is estimated as the ratio between the low and high frequency energy. V_e is applied to evaluate the variance of energy inside a frame. F_v, e_{tilt} and V_e are then compared to predefined thresholds to identify the frame type. The Voiced/Unvoiced classification on the word "success" is illustrated in Fig. 2.
- Other Speech classes: Besides the voiced, unvoiced and silence classifications described above, some specific frames must be distinguished from the voiced/unvoiced frames to avoid speech quality degradation due to time scaling. The average pitch gain of some unvoiced frames is higher than 0.45, while the maximum pitch gain for some voiced frames is below 0.5. These unvoiced and voiced frames are termed as over_voiced and under_voiced frames respectively in this paper. It is suggested not applying time scaling on these frames as well as the plosive frames in order to prevent quality degradation.

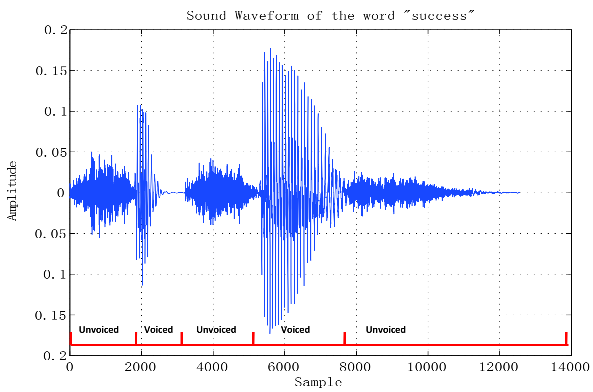

Figure 2. Voiced/Unvoiced classification of the word "success"

B. Time Scale Modification inside AMR decoder

Since pitch lag and pitch gain are internal parameters used by the AMR decoder, it is also advantageous to scale the speech inside the decoder, directly in the excitation domain. According to the speech classification, voiced and unvoiced frames are processed differently. Moreover, some frames are not modified to prevent quality degradation, as proposed in [3]. The different processing operations based on the result of speech classification are summarized in TABLE Ⅱ.

Voiced frames are extended by repeating the pitch cycle preceding the minimum energy point. The number of added pitch cycles is determined by the difference between the original frame length and the expected frame length, combined with the pitch lag of the sub-frame. The voiced frames are suppressed by removing some pitch cycles just before the minimum energy point in the last sub-frame backwards. The number of subtracted cycles depends on the pitch lag of the last sub-frame and the length difference. Time scaling on unvoiced frames is much simpler. A certain number of zeros are uniformly inserted in the unvoiced frame for extension, while zeros are removed from the frame for compression. The number of zeros inserted or removed relies on the expected new length. In unvoiced frames, the samples can be removed from the beginning if the previous frame is unvoiced or from the end of the frame if the previous frame isn't unvoiced. The original and modified signals are illustrated in Fig. 3.

C. Modified AMR decoder

The modified AMR decoder is illustrated in Fig. 4. The generated excitation is formed by the fixed and adaptive codebooks with their corresponding gains. The excitation is classified into voiced/unvoiced/silence and other specific frames. According to the frame type decision, different time scaling techniques are applied to the excitation signal. The reconstructed speech is obtained by feeding the scaled excitation of new length into the LP Synthesis Filter. In order to keep the synchronization between encoder and decoder, the adaptive codebook is updated before time scaling.

TABLE II. TIME SCALING

Frame Type	Time Scale Modification	
	Extension	***Suppression***
Silence	Comfort Noise	Comfort Noise
Voiced	Duplicate some pitch cycles	Remove selected pitch cycles
Unvoiced	Insert zeros between the excitation samples	Remove samples from the excitation signal
Under voiced Over voiced Plosive Onsets	No time scaling	

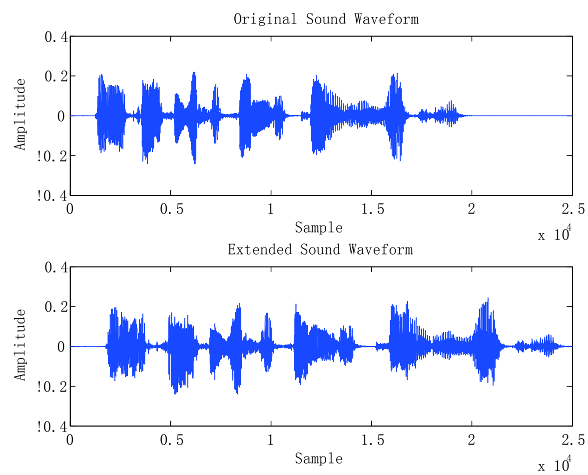

Figure 3. Original and extended sound waveform

V. EXPERIMENTAL RESULTS

In the experiment, we implemented three other most promising algorithms to compare with our proposed jitter buffer algorithm, denoted as Algorithm 1 [1], Algorithm 2 [7] and Algorithm 3 [3]. Our algorithm referred to as Algorithm 4. The results are shown in Table Ⅲ for five traces. Each trace contains 7500 packets and the window size W is set to 300. During the experiment, we implement the proposed play-out delay estimation both on the $I_{e,eff}$ model in (4) and (5). The performance comparison is shown in Fig.5. We observed that both (4) and (5) lead to quite similar results when used in the jitter buffer management algorithm. The maximum length after extension L_{max} is limited to twice of the original length (320 ms) and the minimum length after suppression L_{min} must not be shorter than half the original length (80 ms), as suggested in [3]. The maximum allowable end-to-end delay is 400ms.

From Table Ⅲ, it can be seen that Algorithm 4 achieves the highest MOS scores (which are obtained from the impairment factor R) among all tested traces. Algorithm 1 and Algorithm 2 apply both talk-spurt based jitter buffer management. For Algorithm 1, play-out time is defined with the help of statistical estimation of the play-out delay based on network characteristics of several previous talk-spurts.

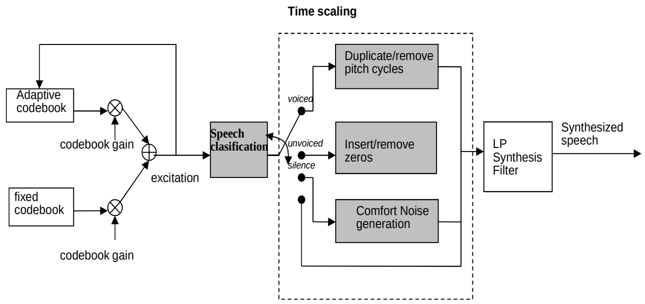

Figure 4. Modified architecture of the AMR decoder

Algorithm 2 implements an extended E-Model for estimation with the assumption of Weibull delay distribution. Both talk-spurt based algorithms are not efficient when the talk-spurts are very long and for cases where the network delay varies significantly such as in cases of spikes. Although a spike detection mechanism is adopted in per-talk-spurt scheduling, the play-out delay cannot be updated until the next talk-spurt. The scheduled play-out time cannot follow such spikes within a talk-spurt and results in more discarded packets due to late arrival, as in trace 1 and trace 5.

Both Algorithm 3 and our proposed algorithm (Algorithm 4) schedule play-out delay on a per-packet basis, thus adjust the play-out time in a highly dynamic way and adapt more quickly to the network conditions even during speech activity (talk-spurt). Algorithm 3 is based on achieving an optimal trade-off between packet loss rate and end-to-end delay, but it does not provide a direct access to the perceived speech quality, which is exactly the goal of the optimization. Our proposed algorithm estimates the play-out delay based on maximizing the MOS value derived from the impairment factor R, therefore achieving best performance in all trace files.

The information of network delay, delay jitter and network loss rate of five trace files are also listed in Table III. The optimal R is calculated by assuming no buffer delay and no late loss rate. The optimal R can only be achieved with full knowledge about the network condition before transmission, thus it cannot be realized in real time QoS monitoring. The difference between the Optimal MOS and the result of our jitter buffer algorithm is partly due to the constraints L_{min} and L_{max} required by time-scaling.

The performance of play-out delay estimation of trace 1, trace 2 are illustrated in Fig.6. Both the results from Algorithm 3 and from our proposed algorithm are shown. Both algorithms adapt play-out delay quite well to the varying network delay. In the cases of spikes, our algorithm reduces the packet loss rate at the expense of additional delay.

VI. Conclusion and Future Work

In this paper, we focused on impairment of random packet loss and end-to-end delay for the AMR codec. We proposed an adaptive jitter buffer algorithm based on the extended E-Model with spike detection and a time scaling technique embedded directly in the AMR decoder. The simulation results show that the proposed method achieve better perceived speech quality compared to other existing algorithms under various network scenarios. Moreover, these results are not specific to AMR. As the time scaling algorithm is closely connected to the CELP coding scheme, the proposed jitter buffer management can be extended to other codecs, in particular to CELP based codecs being most advanced form of speech codecs.

For future activities, subjective listening tests are planned in order to validate the proposed method. We will also extend our work under bursty packet loss conditions.

Acknowledgement

We greatly appreciate the support and guidance of Dr. Herve Taddei, Dr. Christophe Beaugeant and Dr. Imre Varga. We would also like to thank Dr. Anisse Taleb and Mr. David Virette for their helpful comments and advice.

References

[1] S. Moon, J. Kurose, and D. Towsley, "Packet audio play-out delay adjustment: Performance bounds and algorithms," Multimedia Systems, vol.6, no.1, Jan.1998, pp.17-28.

[2] M. Narbutt and L. Murphy, "VoIP Playout Buffer Adjustment using Adaptive Estimation of Network Delays," Proc. 18th Int. Teletraffic Congress (ITC-18), 2003, pp.1171-1180.

[3] Y. Liang, N. Färber, and B. Girod, "Adaptive Play-out Scheduling and Loss Concealment for Voice Communication Over IP Networks," IEEE Trans. on Multimedia, vol.5 2003.

[4] P. Gournay and K. Anderson, "Performance Analysis of a Decoder-Based Time Scaling Algorithm for Variable Jitter Buffering of Speech Over Packet Networks," Proc. IEEE Int. Conf. on Acoustics, May 2006.

[5] K. Fujimoto, S. Ata and M. Murata, "Adaptive playout buffer algorithm for enhancing perceived quality of streaming applications," Proc. IEEE GLOBECOM'02, vol. 3, Nov.2002.

[6] L. Atzori and M. Lobina, "Playout Buffering of Speech Packets Based on a Quality Maximization Approach," IEEE Trans. on Multimedia, vol.8 2006.

[7] L. Sun and E. Ifeachor, "Voice quality prediction models and their application in voip networks," IEEE Trans. on Multimedia, vol.8 2006.

[8] Z. Li, S. Zhao, and J. Kuang, "An improved speech playout buffering algorithm based on a new version of E-Model in VoIP," ChinaCom'08, 2008.

[9] L. Huang, Y. Chen and T. Yaw, "Adaptive VoIP Service QoS Control based on Perceptual Speech Quality," Proc. IEEE ICACT'07, 2007, pp.885-890.

[10] C. Wu and W. Chang, "Perceptual Optimization of Playout Buffer in VoIP Applications, " ChinaCom'06, 2006.

[11] C. Wu, K. Chen, Y. Chang, and C. Lei, "An Empirical Evaluation of VoIP Playout Buffer Dimensioning in Skype, Google Talk, and MSN Messenger," Proc. ACM NOSSDAV'2009, 2009, pp.97-102.

[12] R.Cole and J.Rosenbluth, "Voice over ip performance monitoring, " ACM SIGCOMM, 2001.

[13] ITU-T rec G.107, "E-model, a computational model for use in transmission planning," April 2009.

[14] ITU-T rec G.109, "Definition of categories of speech transmission quality," Sep. 1999.

[15] ITU-T rec G.113, "Transmission impairments due to speech processing," Nov. 2007.

[16] 3GPP TS 26.071, AMR Speech Codec; General description.

[17] 3GPP2 C.S0052-0, Source-Controlled Variable-Rate Multimode Wideband Speech Codec(VMR-WB).

 ISBN: 978-1-61208-127-4

TABLE III. COMPARISON OF DIFFERENT ALGORITHMS AND NETWORK DELAY TRACES

Trace	Algorithm	Average play-out delay(ms)	Late loss rate(%)	MOS	Optimal MOS	Average network delay(ms)	STD of network delay(ms)	Average delay jitter(ms)	Maximum jitter(ms)	Network loss rate (%)
1	Algorithm 1	180.5	7.5	2.0	3.5	136.7	25.0	36.7	146	2.4
	Algorithm 2	191.0	4.0	2.5						
	Algorithm 3	165.2	3.5	2.6						
	Algorithm 4	173.9	2.2	2.9						
2	Algorithm 1	153.3	2.3	3.4	4.1	119.7	12.4	19.7	120	0.24
	Algorithm 2	178.5	0.9	3.8						
	Algorithm 3	150.4	1.5	3.6						
	Algorithm 4	160.0	0.8	3.9						
3	Algorithm 1	148.6	6.0	2.5	4.1	126.8	19.9	26.8	134	0.51
	Algorithm 2	180.6	0.9	3.7						
	Algorithm 3	154.7	1.2	3.7						
	Algorithm 4	158.9	0.7	3.8						
4	Algorithm 1	133.7	0.3	4.1	4.2	112.3	8.8	12.3	48	0
	Algorithm 2	170.0	0.1	4.1						
	Algorithm 3	134.7	0.4	4.1						
	Algorithm 4	134.8	0.3	4.1						
5	Algorithm 1	147.6	2.6	3.4	4.2	116.5	44.9	16.5	305	0
	Algorithm 2	164.4	2.1	3.5						
	Algorithm 3	146.0	1.2	3.8						
	Algorithm 4	148.0	1.0	3.9						

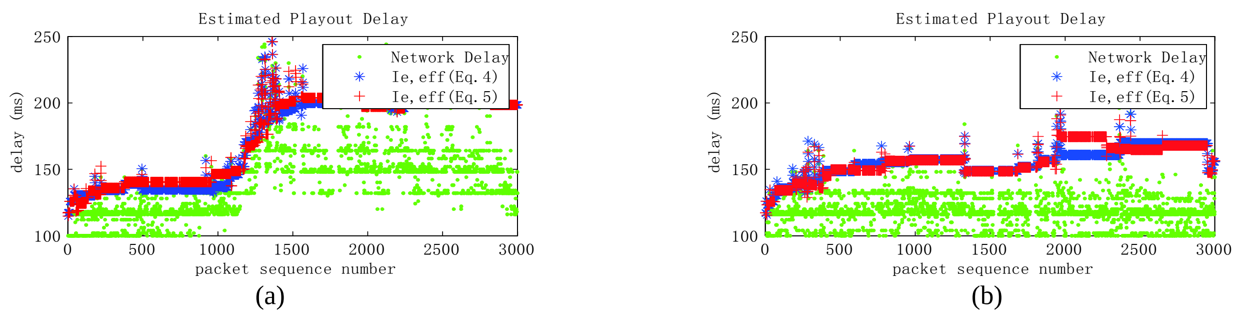

Figure 5. Playout delay estimation based on different $I_{e,eff}$ Models (a) Trace1 (b) Trace2

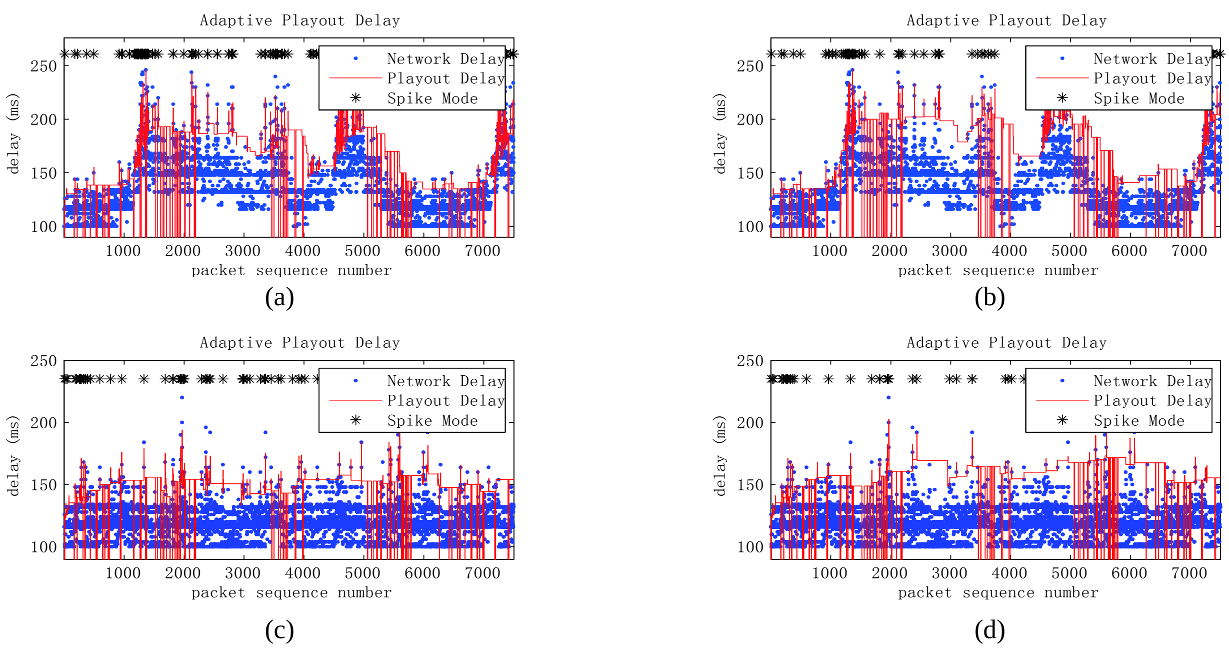

Figure 6. Play-out delay estimation: (a)Algorithm3 for Trace 1 (b)Algorithm4 for Trace1 (c)Algorithm3 for Trace2 (d) Algorithm4 for Trace2