Adaptive Jitter Buffer based on Quality Optimization under Bursty Packet Loss

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*—Quality of voice delivered over packet networks is affected by various factors such as packet loss, end-to-end delay, packet delay variation (jitter) and codec bit rate. Different approaches and models predict speech quality as a function of such impairments. In order to ensure a continuous play-out of voice transmitted over a packet switched network, jitter buffers are commonly used to counter jitter introduced by queuing. In this paper, we propose a new adaptive jitter buffer algorithm based on optimizing the predicted voice quality. The algorithm consists of an adaptive play-out mechanism based on the extended E-Model taking into account packet loss pattern and a time-scaling technique relying on a speech classification mechanism embedded in the decoder. In our work, we apply the time-scaling to the modified AMR WB decoder. Simulation results show that the proposed algorithm outperforms the best existing algorithms in a variety of different network scenarios under bursty packet loss.**

Keywords - adaptive jitter buffer; E-Model; AMR WB; time-scaling; bursty packet loss

I. INTRODUCTION

Transport of Voice over IP (VoIP) is one of the most important applications among recent IP-based telecommunication services. VoIP can be seen as an alternative to the traditional circuit switched telephony with the advantages of reduced cost, simplified network and simplified network management. The main challenge is to guarantee the same Quality of Service (QoS) as that of traditional telephony. Voice quality is the key metric for QoS for VoIP applications. Packet losses, latency (delay) and delay variation (jitter) are the major factors, inevitable in a packet network, contributing to speech quality degradation. In particular, the delay jitter introduced by queuing in packet switched networks has a devastating impact on the perceived quality. In order to smooth delay jitter, a jitter buffer mechanism is required at the receiver for ensuring a continuous play-out of voice data.

When a jitter buffer is applied, received packets are stored in the buffer after arrival, and played out sequentially at scheduled times. Any packet arriving after its scheduled playout time is discarded at the receiver, resulting in the so-called late loss. The late loss rate can be reduced by scheduling a later playout time at the expense of an excessively longer 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 buffers solely based on the trade-off between end-to-end delay (playout delay) and packet loss rate due to late arrival. The playout delay is adjusted either at the beginning of each talk-spurt [2][3] (called per-talk-spurt), or more adaptively within the speech talk-spurt using time-scale modification [4][5] (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, much effort has been devoted to developing approaches to adjust the jitter buffer with the objective of optimizing the perceived speech quality given by the Mean Opinion Score (MOS) [1][6][7][8]. To develop such quality-based approaches, the non-intrusive parametric models which estimate MOS values directly from network parameters and terminal characteristics are required. The ITU-T E-Model [9] is one of the most well-known parametric models. 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 [9].

The ITU-T E-Model has been initially developed as a network planning tool [10]. Although the E-Model has limited accuracy for evaluating conversational speech quality [11][12][13][14], it has shown applicability in the context of QoS monitoring [15][16][17]. In [18][19][20], quality-based playout scheduling approaches were proposed to maximize perceived speech quality using the *R* value of the E-Model as cost function. These approaches adjust the playout delay on per-talk-spurt basis and their performance is limited when talk-spurts are long and the network delay varies significantly. A per-packet quality-based jitter buffer algorithm is described in [21]. The playout delay estimation is designed as an unconstrained optimization problem that maximizes the *R* value. However, in per-packet jitter buffer management, a speech frame can only be time-scaled within a certain range to maintain the naturalness of the original speech signal [4]. Thus, a constrained optimization problem is more suitable.

To design a quality-based jitter buffer algorithm, 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, such as Pareto [21], Weibull [18] and Gamma [22]. 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 particular for jitter buffer

management on per-packet basis, the network delay behavior cannot be modeled just by a certain type of distribution.

Many previous works focus on the packet loss impairment [11][12][13][23][24]. Some studies assume random packet loss for quality estimation and focus only on the overall packet loss rate [18][25]. Several studies revealed temporal dependencies in packet loss based on network statistics. The overall loss rate alone is not sufficient to predict the speech quality perceived by users [26]. The authors proposed a method to calculate the burstiness level from the packet loss pattern which can be converted into an equivalent random packet loss factor [27]. The results in [28] have shown that other properties such as the loss location and the loss distribution also have impact on the perceived speech quality.

In [1], we presented an adaptive jitter buffer system implementing per-packet scheduling based on the extended E-Model. The system contains a spike detection mechanism and a classifier based time-scaling technique similar to that proposed in [5]. The time-scaling technique is implemented directly inside the *AMR-NB* decoder [29]. It is advantageous for the quality and makes it possible to use the internal parameters such as the pitch lag and the gains for time-scaling. In this paper, we extend our previous work by taking into account bursty packet loss and adapting the mechanism to wideband speech transmission for *AMR-WB* [30][31].

The rest of the paper is organized as follows. Section II gives a brief overview of the extended E-Models including the impairment model for random packet loss and bursty packet loss. In Section III, the play-out algorithm based on optimizing the predicted voice quality is proposed. Section IV presents the modified time-scaling embedded in the *AMR-WB* decoder. Simulation results illustrating the performance of the proposed scheme are presented in Section V. Finally, we conclude this paper in Section VI.

II. EXTENDED E-MODEL

A. ITU-T E-Model

The ITU-T E-Model is a computational model for the prediction of the expected voice quality which combines different impairments contributing to speech quality degradation, such as loudness, background noise, low bit-rate coding distortion, codec, echo, packet loss and delay. A large set of these impairment factors have been quantified regarding their impact on the conversational speech quality. The underlying assumption of the E-Model is that all those impairment factors 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. An invertible mapping exists between *R* and conversational MOS. A rating of '0' represents a MOS value '1' (bad quality) and '100' of *R* represents MOS value '4.5' (high quality). For wideband speech transmission, *R* can go beyond 100. The output *R* value is obtained by subtracting impairment factors from a basic quality measure [9]:

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

where R_0 represents the basic signal-to-noise ratio; I_s is the Simultaneous Impairment Factor which occurs more or less simultaneously with the speech signal; I_d represents the impairments caused by delay; $I_{e,eff}$ is the Effective Equipment Impairment Factor representing impairments caused by low bit rate codecs and packet loss. A is an Advantage Factor which has accordingly no relationship to any other parameter and normally can be neglected. $I_{e,eff}$ and I_d are the most important factors to predict voice quality in packet networks.

B. Effective Equipment Impairment Factor

The model of $I_{e,eff}$ in [18] is defined as

$$I_{e,eff} = I_e + (95 - I_e) \cdot \frac{P_{pl}}{\frac{P_{pl}}{BurstR} + B_{pl}} \quad (2)$$

where I_e is a codec specific value and represents the equipment factor in loss-free networks. B_{pl} represents the robustness of the codec in lossy networks. ITU-T G.113 Appendix [32] lists the provisional reference values of several codecs, derived from subjective MOS test results and network experience. Both the packet loss probability P_{pl} and the Burst Ratio $BurstR$ depend on the packet loss pattern. If $BurstR$ = 1, the packet loss is random. If $BurstR$ >1, the packet loss is bursty. Otherwise, the packet loss distribution is less bursty. Bursty packet loss can be emulated by a two state Markov model characterized by two transition probabilities: p between "Found" and "Loss" state, and q between "Loss" and "Found" state. p and q can be obtained from the mean loss rate (mlr) and the mean burst length (mbl) as

$$p = \frac{mlr}{mbl(1 - mlr)}, q = \frac{1}{mbl} \quad (3)$$

$BurstR$ can be calculated as [18]

$$BurstR = \frac{1}{p + q} \quad (4)$$

By combining (3) and (4), $BurstR$ is calculated as

$$BurstR = mbl(1 - mlr) \quad (5)$$

As $P_{pl} = mlr \cdot 100$, we can express $BurstR$ as a function of mbl and P_{pl}

$$BurstR = \left(1 - \frac{P_{pl}}{100}\right) \cdot mbl \quad (6)$$

As mbl represents the average length of burst in an arrival sequence, it can be calculated by counting the number of consecutively lost packets and multiplying the count with the corresponding probability as

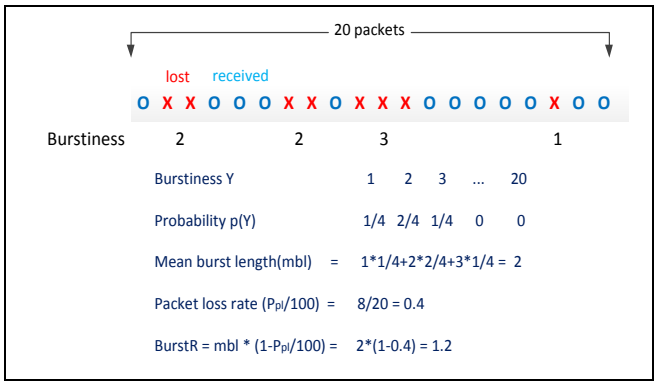


Figure 1. Example of *BurstR* Calculation

TABLE I. PROVISIONAL VALUE

Codec	I_e	B_{pl}
AMR-WB 12.65 kbit/s	20	4.3
AMR-NB 12.2 kbit/s	5	10

$$mbl = \sum_{k=0}^{\infty} kP(Y=k) \quad (7)$$

where $P(Y=k)$ is the probability for having k consecutively lost packets. With (6) and (7), we can obtain $BurstR$ as

$$BurstR = \left(1-\frac{P_{pl}}{100}\right)\sum_{k=0}^{\infty} kP(Y=k) \quad (8)$$

Fig. 1 gives an example of how to calculate the $BurstR$ for a particular sequence of packets. The overall number of packets is assumed to be 20. The packet loss pattern is described by color in Fig. 1: the red X means a lost frame and the blue O represents a successfully received frame. By giving the sequence of error information, we can calculate an instantaneous $BurstR$ as 1.2 for this sequence, as shown in Fig. 1.

We further divide P_{pl} into two parts

$$P_{pl} = 100 \cdot (p_n + p_b) \quad (9)$$

where p_n is the packet loss rate in the network and p_b is the late packet loss rate caused by packet drops in the jitter buffer. Since a packet is discarded when it arrives after its scheduled playout time, the late loss rate p_b is calculated as

$$\begin{aligned} p_b &= (1-p_n)\big(1-P_r(X \le d)\big) \\ &= (1-p_n)\big(1-F(d)\big) \end{aligned} \quad (10)$$

with $F(d)$ being the CDF of network delay (d) obtained from histogram statistics of previous network delays.

The provisional planning values of I_e and B_{pl} required in (2) can be found in [32][33], and the values of *AMR-NB/AMR-WB* are listed in Table I. Applying these values to (2), we can calculate the $I_{e,eff}$ for *AMR-NB* 12.2 kbit/s as

$$\begin{aligned} I_{e,eff} &= 5 + 90\frac{P_{pl}}{\frac{P_{pl}}{BurstR}+10} \\ &= 5 + 90\frac{p_n+p_b}{\frac{p_n+p_b}{BurstR}+\frac{10}{100}} \end{aligned} \quad (11)$$

However, the provisional reference values have to be derived from a large quantity of subjective tests. In order to avoid subjective tests, alternatively, $I_{e,eff}$ can be estimated with a logarithmic fitness curve as given in [23][24]:

$$I_{e,eff} = A + B\,ln(1 + C(p_n + p_b)) \quad (12)$$

where *A*, *B* and *C* are curve fitting parameters. Other codecs may have different forms of curve for $I_{e,eff}$ [34]. The empirical formula for *AMR-NB* 12.2 kbit/s is [18]

$$I_{e,eff} = 14.96 + 16.68\,ln(1 + 30.11(p_n + p_b)) \quad (13)$$

A similar equation to (2) for the wideband effective equipment factor $I_{e,eff,WB}$ under random packet loss ($BurstR = 1$) is given by [9][35][36]. Similarly, for wideband speech codec such as *AMR-WB* 12.65 kbit/s, $I_{e,eff,WB}$ can be obtained by applying provisioning values from [33] for random packet loss

$$\begin{aligned} I_{e,eff,WB} &= 20 + 75\frac{P_{pl}}{P_{pl}+4.3} \\ &= 20 + 75\frac{p_n+p_b}{p_n+p_b+\frac{4.3}{100}} \end{aligned} \quad (14)$$

For bursty packet loss, an empirical formula for $I_{e,eff,WB}$ was proposed in [13][26] using Genetic Programming for different wideband codecs. For example, the $I_{e,eff,WB}$ for *AMR-WB* 12.65 kbit/s is expressed as

$$\begin{aligned} I_{e,eff,WB} = \Big\{ &\ln\left(\frac{9\cdot(43.91+P_{pl}\cdot 187.62^2)}{mbl^5-P_{pl}}\right) + P_{pl} + 43.91 + \\ &187.62\cdot P_{pl}\Big\}\cdot 0.8303 + 8.9977 \end{aligned} \quad (15)$$

C. Delay Impairment Factor

If I_d refers to impairments only due to end-to-end delay d, then I_d can be derived by curve fitting as described in [25]

$$I_d = 0.024d + 0.11(d-177.3)H(d-177.3) \quad (16)$$

where $H(x)$ is the step function ($H(x)=0$ if $x<0$; $H(x)=1$ else).

D. Extended E-Model

For the extended E-Model, we assume that impairments due to other factors such as echo or delay are not present. All input parameters and their recommended ranges are found in [9]. For those parameters which are not available at the time of planning, the default values from the ITU [32][37] are recommended. We only focus on IP networks, and the expression of E-Model in (1) can be simplified in terms of transport-level metrics [25] for narrow band

$$R_{NB} = 93.2 - I_d - I_{e,eff} \quad (17)$$

And a similar expression for wide band [25] is

$$R_{WB} = 129 - I_d - I_{e,eff,WB} \quad (18)$$

If we define the sum of I_d and $I_{e,eff}$ as a new impairment factor I

$$I = I_d + I_{e,eff} \quad (19)$$

then both (17) and (18) can be simplified as

$$R = R_{max} - I \quad (20)$$

where R_{max} is 93.2 for narrow band and 129 for wide band.

This formulation of R in (20) is used as the cost function in our jitter buffer management to estimate the playout delay by maximizing R which is equivalent to minimizing I. Equation (11) with $BurstR$ = 1 is already used in [1] for *AMR-NB* random packet loss. We further investigate (11) for modeling $I_{e,eff}$ of *AMR-NB* under bursty packet loss, (14) for modeling $I_{e,eff,WB}$ of *AMR-WB* under random packet loss, and also the empirical formula (15) for modeling $I_{e,eff,WB}$ of *AMR-WB* under bursty packet loss. The results are shown in Section V.

III. Proposed Play-out Algorithm

The proposed receiver includes an adaptive jitter buffer algorithm and a time-scaling embedded inside the decoder, as shown in Fig. 2. The adaptive play-out algorithm is the main control unit. Since spikes are very common in VoIP transmission, spike detection in [3] is implemented to switch between NORMAL mode and SPIKE mode. In SPIKE mode, the scheduled playout time follows current network condition. In NORMAL mode, the scheduled playout time is estimated based on the extended E-Model, as discussed in Section II.

The play-out algorithm is similar to the one proposed in [1], and will be described using the same basic notations listed in Table II. For each packet, a certain playout time is scheduled at the receiver before its arrival. 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 playout delay of the next expected packet has to be estimated to obtain the expected frame length of the current frame. The playout delay is chosen in order to maximize the predicted 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, late loss rate p_b and also $BurstR$ regarding bursty packet loss. Both network loss rate and $BurstR$ can be calculated based on the loss pattern of previous received packets stored in a history window with the window size W. The late loss rate is determined by the playout buffering algorithm, and thus by the end-to-end delay (playout delay). Therefore, (20) can be expressed as a function of playout delay, and applied as the cost function in the playout buffering algorithm to predict the voice quality.

The playout 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 $packet^i$, and obtain network delay information d_n^i and error information from the RTP header. The loss pattern of the most recent received W (history window size) packets is updated and $BurstR$ is calculated (default is 1).
2. Spike detection: check the current network condition, and switch between SPIKE/ NORMAL.
3. Playout 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 playout delay
      $d_p^i = \hat{d}_p^i$
4. Playout delay estimation
   a) SPIKE: follow the current network delay $\hat{d}_p^{i+1} = d_n^i$, and skip step 5.
   b) NORMAL: estimate playout delay based on the E-Model.
5. E-Model based playout 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 playout delay for $packet^{i+1}$
      $\hat{d}_p^{i+1}: I_m(\hat{d}_p^{i+1}) = \min_{d_{min} \le d \le d_{max}} I_m(d)$
      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.

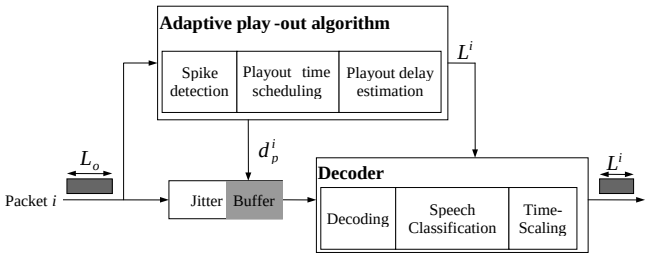


Figure 2. Proposed adaptive jitter buffer at the receiver

TABLE II. BASIC NOTATIONS

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

IV. TIME-SCALING EMBEDDED IN THE DECODER

The E-Model based playout scheduling algorithm described in Section III is applied specifically to the CELP codec. The standard 3GPP *AMR-WB* decoder [30] is modified to embed the time-scaling technique based on speech classification. According to the evaluated frame type, different time-scaling (extension or suppression) operations are applied to the frame in the excitation domain.

A. Speech Classification

In our previous work [1], the speech classification was based on three parameters: the Voicing Factor, the Spectral Ratio and the Energy Variation. The frame type is thus identified by comparing them to the predefined reference values. Special frames such as plosive or over-voiced frames are also differentiated from others by using the internal parameters inside the *AMR-NB* decoder. In order to make the classification more accurate, we implement the merit function f_m which has been defined in the *VMR* codec [38] for the frame erasure concealment as

$$f_m = \frac{1}{7}\left(2\bar{R}_{xy}^s + e_{tilt}^s + SNR^s + pc^s + E_{rel}^s + zc^s\right) \qquad (21)$$

where $[\cdot]^s$ represents the scaled version of the corresponding classification parameters, including: the normalized correlations $\bar{R}_{xy}$, the spectral tilt parameter e_{tilt} estimated as the ratio between the low and high frequency energy, the signal to noise ratio SNR of the current frame, the pitch stability counter pc representing the pitch period variation, the relative frame energy E_{rel} and the zero-crossing counter zc inside a frame. These parameters are considered together to build the merit function. The classification decision of the current frame is made depending on the value of f_m and also on the previous frame type.

The classification rules and also the calculation of all the parameters are thoroughly explained in [38]. Following the rules, the current frame is classified as VOICED, UNVOICED, ONSET and other TRANSITION frames. The silence frame is classified as UNVOICED here. The silence frame is identified by checking the VAD flag if operated in DTX mode, otherwise, the merit function is used. VOICED contains stable and periodic components. UNVOICED includes silence frame and is more like white noise. TRANSITION (including VOICED TRANSITION and UNVOICED TRANSITION) and ONSET are characterized by rapid variations of the energy. The speech classification on the word "success" is illustrated in Fig. 3.

B. Time-scaling in the excitation domain

The stretch and suppress operation for a speech sequence are illustrated in Fig. 4 (a) and Fig. 4 (b), respectively. The speech sequence can comprise VOICED, UNVOICED, TRANSITION and also ONSET frames, as illustrated in Fig. 3. According to the speech classification of each frame, VOICED and UNVOICED frames are processed differently. Moreover, some frames are not modified to prevent quality degradation, as proposed in [4]. Since pitch lag and pitch gain are internal parameters used by the *AMR* decoders, it is also advantageous to scale the speech inside the decoder, directly in the excitation domain. The different processing operations based on the result of speech classification are summarized as follows.

1) Time-scaling on VOICED: VOICED frames are extended or suppressed by adding or removing a certain number of pitch cycles to preserve the periodic nature. In the loop for each subframe in the analysis frame, information of pitch lags and pitch gains is first decoded, and also the adaptive codebook is updated before time-scaling to keep synchronization. The number of added or subtracted pitch cycles is determined by the difference between the original frame length and the expected frame length, combined with the pitch lag of the subframe. For extension, a certain number of pitch cycles are added just before the minimum energy point in the excitation signal, as shown in Fig. 5 (a). In the subframe with the maximum pitch gain, a search window of 20 samples is used to identify the minimum energy point P_{min}. For suppression, a certain number of pitch cycles are removed just before the minimum energy point P_{min} backwards, as shown in Fig. 5 (b). The minimum energy point P_{min} is found by searching in the last subframe. Fig. 6 (a) and Fig. 6 (b) show the extension and suppression of the same VOICED frame in the synthesis domain.

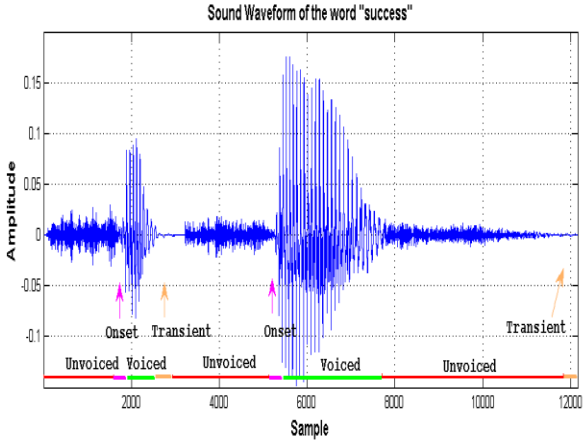


Figure 3. Speech classification of the word "success"

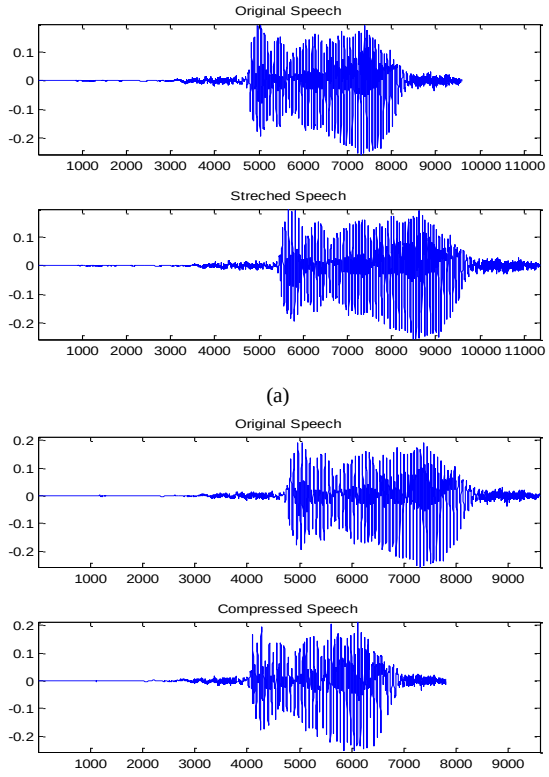


Figure 4. Time-scaling for speech sequence: (a) The speech sequence is streched. (b) The speech sequence is compressed

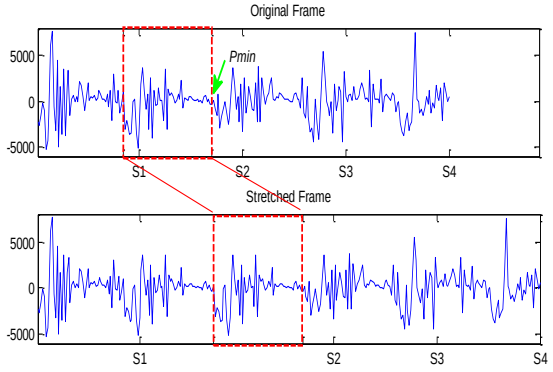
2) Time-scaling on UNVOICED: Time-scaling on UNVOICED frames is much simpler. For extension, a certain number of zeros are uniformly inserted in the excitation of the frame. In order to maintain the average energy per sample, a weighting factor which is the ratio between the requested and original frame lengths, is multiplied to the time-scaled excitation signal. For suppression, a certain number of samples are removed from the excitation of the frame. The samples can be removed from the beginning of the frame if the previous frame is UNVOICED or from the end of the frame if the previous frame is of other types. The number of zeros inserted or the number of samples removed relies on the expected new length, the original length and also the pitch lag.

3) Time-scaling on TRANSITION and ONSET: Since TRANSITION and ONSET frames contain components characterized by rapid variations, no time scale modification is operated on these frames to avoid artifacts.

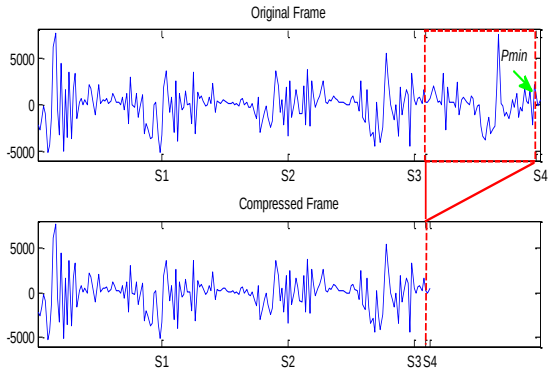
C. Modified CELP decoder

The modification of a simplified CELP synthesis decoder is illustrated in Fig. 7. Although the details can be different in each specific codec, the basic idea is quite similar. The generated excitation is formed by the fixed and adaptive codebooks with their corresponding gains. The excitation is classified into VOICED/UNVOICED/ONSET and other TRANSITION 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.

In the *AMR-WB* decoder, the modification is a little bit more complicated, as the synthesis signal comprises two parts: the low-band synthesis SYN_{lo} and the high-band synthesis SYN_{hi}. In the loop of each subframe of the original length $L_{o,sub}$ (64 samples), the low-band excitation EXC_{lo} is formed by the fixed and adaptive codebooks with corresponding gains. Then the EXC_{lo} is post-processed and time scaled depending on the speech characteristics. The scaled low-band excitation is of the new length $L_{12.8,sub}$. The low-band synthesis SYN_{lo} is obtained by feeding the low-band excitation signal with the new excitation of length $L_{12.8,sub}$ to the synthesis filter and then performing up-sampling. Then the new low-band synthesis is of the new length $L_{16,sub} = L_{12.8,sub} \cdot 5/4$. The high-band synthesis is generated as usual but scaled with the new length $L_{16,sub}$. Finally, the high-band synthesis SYN_{hi} is added to the low-band synthesis SYN_{lo} to produce the synthesized speech signal SYN of the new length $L_{16,sub}$.

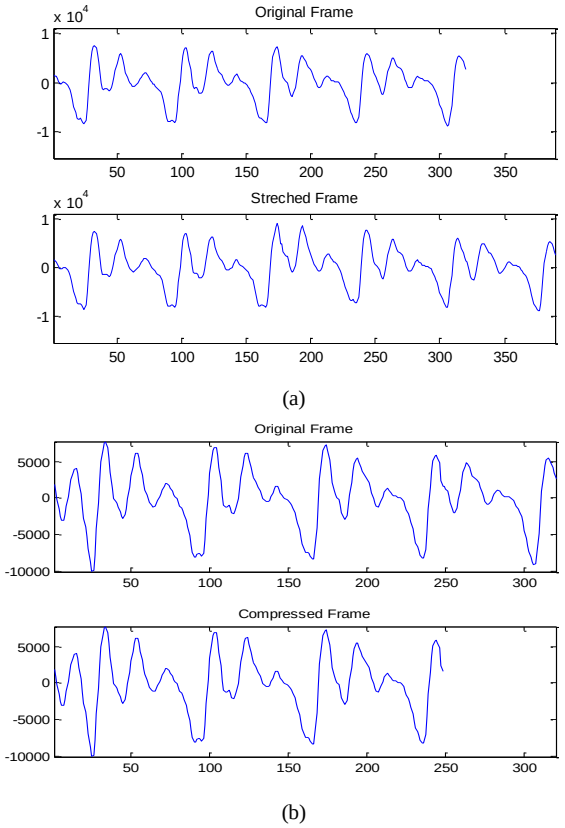


(a)



(b)

Figure 5. Time-scaling for VOICED frame in the excitation domain: (a) The frame is streched by adding one pitch cycle (b) The frame is compressed by removing one pitch cycle.



(a)

(b)

Figure 6. Time-scaling for VOICED frame in the synthesis domain: (a) The frame is streched by adding one pitch cycle (b) The frame is compressed by removing one pitch cycle.

V. EXPERIMENTAL RESULTS

In the experiment, we implemented three other most promising algorithms, denoted as Algorithm 1 [2], Algorithm 2 [18], Algorithm 3 [4] to compare with our proposed jitter buffer algorithm. Our algorithm in this paper refers to Algorithm 4 for *AMR-NB* under random loss [1], Algorithm 5 for *AMR-NB* under bursty loss, Algorithm 6 for *AMR-WB* under random loss and Algorithm 7 for *AMR-WB* under bursty packet loss. The five traces used in the simulation are the same as used in [1]: each trace contains 7500 packets and the window size W used for both burstiness history and delay history is set to 300. The network statistics information for all five traces is shown in Table III. During the experiment, we implement the proposed playout delay estimation on the $I_{e,eff}$ model in (11) for Algorithm 5, (14) for Algorithm 6 and (15) for Algorithm 7. The maximum length after extension L_{max} is limited to twice of the original length (320 samples) and the minimum length after suppression L_{min} must not be shorter than half the original length (80 samples), as suggested in [4]. The maximum allowable end-to-end delay is 400 ms. The results for *AMR-NB* 12.2 kbit/s and *AMR-WB* 12.65 kbit/s are shown in Table IV.

From Table IV, it can be seen that Algorithm 5 achieves the highest R scores (consequently the highest MOS – see Section II) for *AMR-NB* and Algorithm 7 achieves the highest R scores for *AMR-WB* for all the tested traces. However, for narrow band Algorithm 4 has similar performance as Algorithm 5, and also for wide band the difference between Algorithm 6 and Algorithm 7 is minor. The performance comparison between Algorithm 4 and Algorithm 5 is shown in Fig. 8. We only show the results for trace 1 and trace 3 here, since they have the highest loss rate among all the trace files, as illustrated in Table III. The random loss model (Algorithm 4) and the bursty loss model (Algorithm 5) can lead to quite similar results when no packet or only a few packets are lost, i.e., $R \approx 1$. When the loss density is high (bursty), i.e., $BurstR > 1$, the difference can be perceived.

Algorithm 1 and Algorithm 2 are both talk-spurt based algorithms. Algorithm 1 estimates the playout time with the help of statistical delay information of several previous talk-spurts. Algorithm 2 implements an extended E-Model based on Weibull delay distribution. Both algorithms are not efficient for long talk-spurts and for cases where the network delay varies significantly such as in cases of spikes. Since the playout delay can only be updated in the next talk-spurt, the scheduled playout time cannot follow such spikes within a talk-spurt and results in more discarded packets due to late arrival, as for trace 1 and trace 5. Algorithm 3 and Algorithms 4 to 7 schedule playout delay on a per-packet basis. Algorithm 3 adjusts the playout time based on achieving an optimal trade-off between packet loss rate and end-to-end delay in a highly dynamic way and adapts more quickly to the network conditions even during speech activity (talk-spurt). But it does not provide a direct influence on the perceived speech quality, which is exactly the goal of the optimization. Algorithms 4 to 7 estimate the playout delay based on maximizing the MOS value derived from the rating factor R, therefore achieving the best performance in all the trace files.

The performance of playout delay estimation of trace 1 and trace 2 for *AMR-WB* is illustrated in Fig. 9. The results from Algorithm 3 and from our proposed Algorithm 6 are shown. Both algorithms adapt the playout 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.

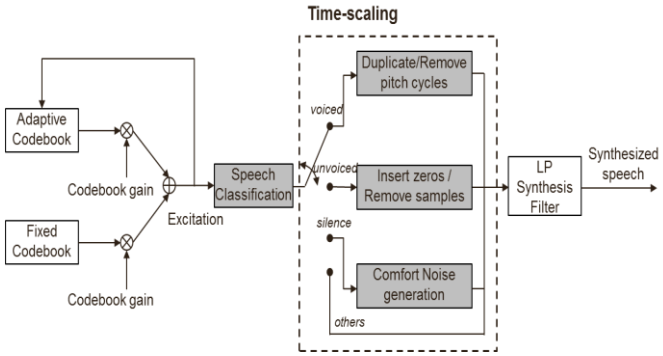


Figure 7. Modified architecture of the CELP synthesis model

TABLE III. TRACE FILE STATISTICS

Trace	Average network delay(ms)	STD of network delay(ms)	Average delay jitter(ms)	Maximum jitter(ms)	Network loss rate (%)
1	136.7	25.0	36.7	146	2.4
2	119.7	12.4	19.7	120	0.24
3	126.8	19.9	26.8	134	0.51
4	112.3	8.8	12.3	48	0
5	116.5	44.9	16.5	305	0

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed an adaptive jitter buffer algorithm based on the extended E-Model under bursty packet loss. We focused on the *AMR* codecs and a time-scaling technique embedded in the *AMR* decoders. Although the E-Model has limited accuracy in evaluating the speech quality, it can be used well in voice quality monitoring. Our simulation results show that the proposed method achieves better perceived speech quality compared to other existing algorithms under various network scenarios. Moreover, these results are not specific to *AMR-NB* and *AMR-WB*. As the time-scaling algorithm is closely related to the CELP coding scheme, the proposed jitter buffer management can be extended and adapted to other codecs, in particular to CELP based codecs.

For future activities, subjective listening tests are planned in order to validate the proposed method.

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] L. Pang and L. Böszörmenyi, "E-Model based Adaptive Jitter Buffer with Time-Scaling Embedded in AMR decoder," Proc. Int. Conf. on Digital Telecommunications, ICDT'11, April 2011, pp. 80-85.

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

[3] 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.

[4] 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.

[5] 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, Speech and Signal Processing, ICASSP'06, May 2006.

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

[7] L. Atzori and M. Lobina, "Speech playout buffering based on a simplified version of the ITU-T E-model," IEEE Signal Processing Letters, vol. 11, no. 3, March 2004, pp. 382-385.

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

[9] ITU-T rec G.107, "E-model, a computational model for use in transmission planning," Dec. 2011.

[10] N.O. Johannesson, "The ETSI computation model: a tool for transmission planning of telephone networks," IEEE Communications Magazine, vol. 35, no. 1, Jan. 1997, pp. 70-79.

[11] A. Rix, J.G. Beerends, D. Kim, P. Kroon, and O. Ghitza, "Objective assessment of speech and audio quality — Technology and applications," IEEE Trans. on Audio, Speech and Language Processing, vol. 14, 2006, pp. 1890-1901.

[12] A. Raja, R. Azad, C. Flanagan, and C. Ryan, "VoIP Speech Quality Estimation in a Mixed Context with Genetic Programming," Proc. 10th Annual Conf. on Genetic and evolutionary computation, July 2008, pp. 1627-1634.

[13] A. Raja, R. Azad, C. Flanagan, and C. Ryan, "A Methodology for Deriving VoIP Equipment Impairment Factors for a Mixed NB/WB Context," IEEE Trans. on Multimedia, vol. 10, no. 6, pp. 1046-1058, Oct. 2008.

[14] L. Carvalho, et al., "An E-Model Implementation for Speech Quality Evaluation in VoIP Systems," Proc. IEEE Symposium Computers and Communications, ISCC'05, 2005, pp. 933-938.

[15] S. Paulsen and T. Uhl, "Adjustments for QoS of VoIP in the E-Model," Telecommunications: The Infrastructure for the 21st Century, WTC, Sept. 13-14, 2010, pp. 1-6.

[16] W. Chen, P. Lin, and Y. Lin, "Real-Time VoIP Quality Measurement for Mobile Devices," IEEE Systems Journal, vol. 5, no. 4, Dec. 2011, pp. 538-544.

[17] K.S. Shanmugan, "Simulation-Based Estimate of QoS for Voice Traffic over WCDMA Radio Links," Proc. 5th Int. Conf. on Wireless Communications, Networking and Mobile Computing, WiCom'09, Sept. 24-26, 2009, pp. 1-4.

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

[19] Z. Li, S. Zhao, and J. Kuang, "An improved speech playout buffering algorithm based on a new version of E-Model in VoIP," Proc. 3rd Int. Conf. on Communications and Networking in China, ChinaCom'08, 2008.

[20] L. Huang, Y. Chen, and T. Yaw, "Adaptive VoIP Service QoS Control based on Perceptual Speech Quality," Proc. 9th Int. Conf. on Advanced Communication Technology, 2007, pp. 885-890.

[21] C. Wu and W. Chang, "Perceptual Optimization of Playout Buffer in VoIP Applications," Proc. 1st Int. Conf. on

Communications and Networking in China, ChinaCom'06, 2006.

[22] C. Wu, K. Chen, Y. Chang, and C. Lei, "An Empirical Evaluation of VoIP Playout Buffer Dimensioning in Skype, Google Talk, and MSN Messenger," 18th Int. Workshop on Network and operating systems support for digital audio and video, 2009, pp. 97-102.

[23] L. Sun, G. Wade, B. Lines, and E. Ifeachor, "Impact of Packet Loss Location on Perceived Speech Quality," 2nd IP-Telephony Workshop, IPTEL'01, April 2001, pp. 114-122.

[24] L. Sun and E. Ifeachor, "New models for perceived voice quality prediction and their applications in playout buffer optimization for VoIP networks," Proc. IEEE Int. Conf. on Communications, ICC'04, vol. 3, June 20-24, 2004, pp. 1478-1483.

[25] R. Cole and J. Rosenbluth, "Voice over ip performance monitoring," ACM SIGCOMM Computer Communication Review, 2001.

[26] A. Raake, "Short- and long-term packet loss behavior: Towards speech quality prediction for arbitrary loss distributions," IEEE Trans. on Audio, Speech, Language Processing, vol. 14, no. 6, Nov. 2006, pp. 1957-1968.

[27] H. Zhang, L. Xie, J. Byun, P. Flynn, and C. Shim, "Packet loss burstiness and enhancement to the E-Model," Proc. 6th ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD'05, May 23-25, 2005, pp. 214-219.

[28] J. Liu and G. Wei, "Model the Packet-Loss Dependent Effective Equipment Impairment Factor in Speech Quality Estimation in VoIP and Its Realization," Proc. 2nd Int. Conf. on Advanced Computer Control, ICACC'11, vol. 4, March 27-29, 2011, pp. 359-362.

[29] 3GPP TS 26.071, AMR Speech Codec; General description, March 2011.

[30] 3GPP TS 26.171, AMR Wideband Speech Codec; General description, March 2011.

[31] B. Bessette, et al., "The adaptive multirate wideband speech codec (AMR-WB)," IEEE Trans. on Speech and Audio Processing, vol. 10, no. 8, Nov. 2002, pp. 620-636.

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

[33] ITU-T rec G.113 Amendment 1, "Revised Appendix IV-Provisional planning values for the wideband equipment impairment factor and the wideband packet loss robustness factor," March 2009.

[34] M. Goudarzi, L. Sun, and E. Ifeachor, "Modelling Speech Quality for NB and WB SILK Codec for VoIP Applications," Proc. 5th Int. Conf. on Next Generation Mobile Applications, Services and Technologies, NGMAST'11, Sept. 14-16, 2011, pp. 42-47.

[35] S. Möller, A. Raake, N. Kitawaki, and A. Takahashi, "Impairment factor framework for wideband speech codecs," IEEE Trans. on Audio, Speech, Language Processing, vol. 14, no. 6, Nov. 2006, pp. 1969-1976.

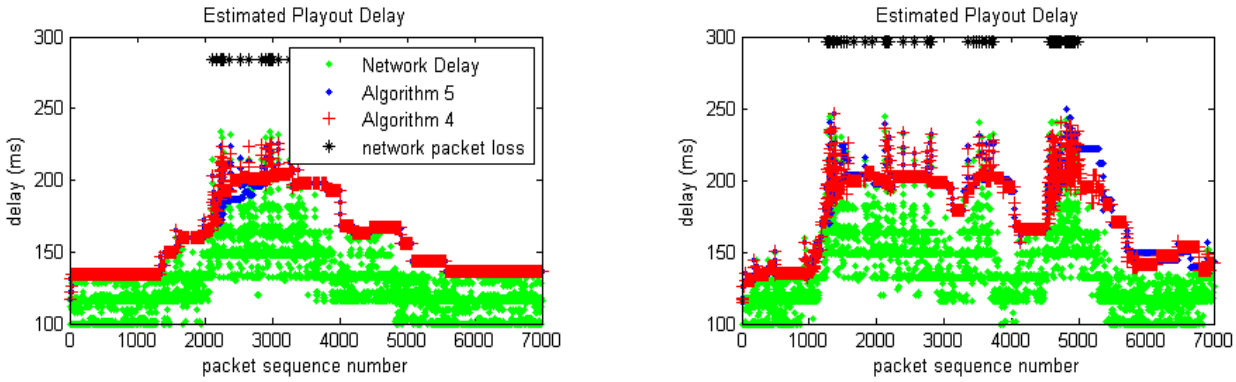
[36] S. Möller, N. Côté, V. Gautier-Turbin, N. Kitawaki, and A. Takahashi, "Instrumental Estimation of E-Model Parameters for Wideband Speech Codecs," EURASIP Journal on Audio, Speech and Music Processing, 2010.

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

[38] 3GPP2 C.S0052-0, Source-Controlled Variable-Rate Multimode Wideband Speech Codec (VMR-WB), June 2004.

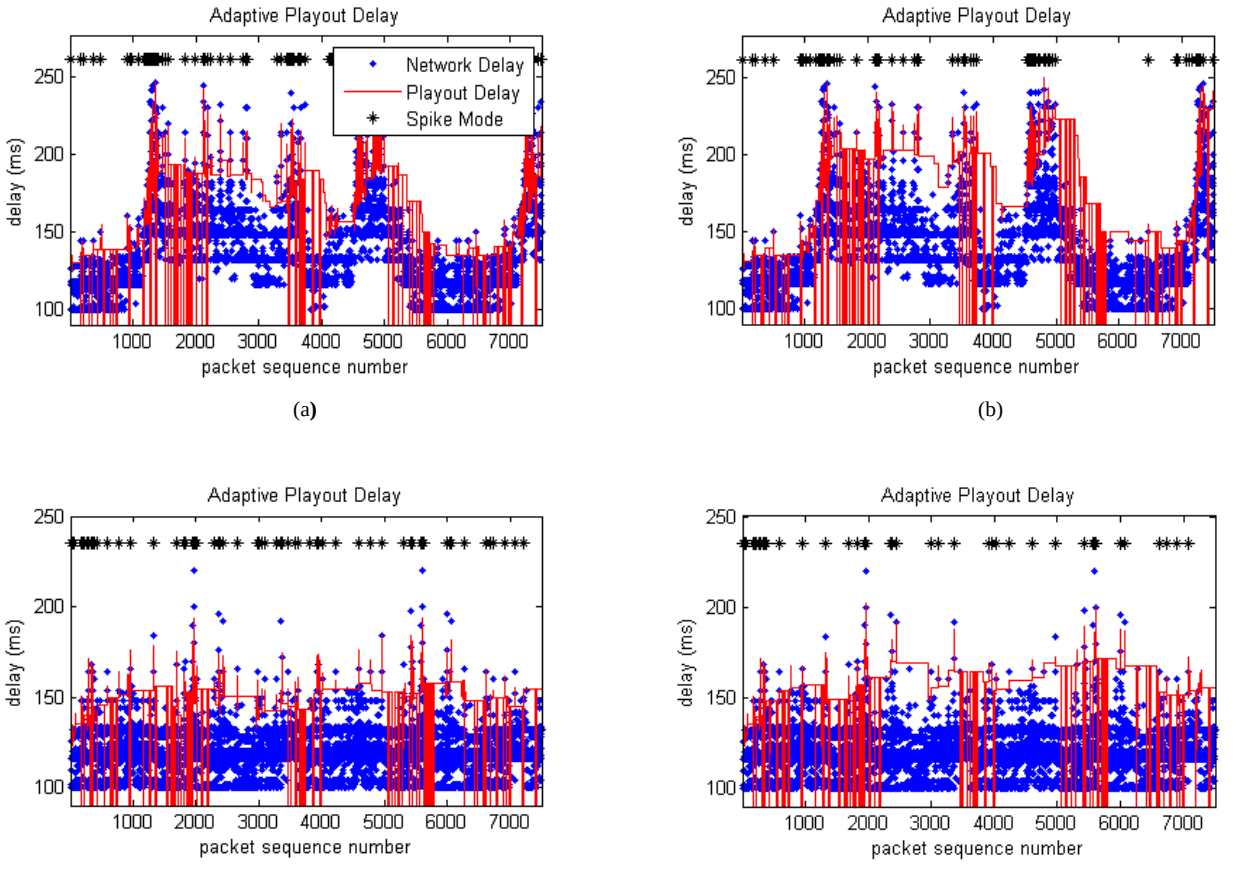
TABLE IV. COMPARISON OF DIFFERENT ALGORITHMS

	AMR-NB 12.2 kbit/s				AMR-WB 12.65 kbit/s			
Trace	Algorithm	Average playout delay(ms)	Late loss rate(%)	R	Algorithm	Average playout delay(ms)	Late loss rate(%)	R
1	Algorithm 1	180.5	7.5	39	Algorithm 1	180.5	7.5	40.5
	Algorithm 2	191.0	4.0	47.7	Algorithm 2	190.9	3.9	45.5
	Algorithm 3	165.2	3.5	51.3	Algorithm 3	165.2	3.5	48.1
	Algorithm 4	173.9	2.2	57.3	Algorithm 6	177.1	2.0	52.0
	Algorithm 5	175.9	1.9	57.3	Algorithm 7	176.8	2.1	52.1
2	Algorithm 1	153.3	2.3	66.1	Algorithm 1	153.3	2.3	59.9
	Algorithm 2	178.5	0.9	75.0	Algorithm 2	178.5	0.9	69.4
	Algorithm 3	150.4	1.5	71.0	Algorithm 3	150.4	1.5	64.7
	Algorithm 4	160.0	0.8	75.9	Algorithm 6	159.7	0.8	70.2
	Algorithm 5	160.0	0.8	75.9	Algorithm 7	159.7	0.8	70.2
3	Algorithm 1	148.6	6.0	49.2	Algorithm 1	148.6	5.0	46.8
	Algorithm 2	180.6	0.9	72.2	Algorithm 2	180.6	0.9	66.3
	Algorithm 3	154.7	1.2	71.5	Algorithm 3	154.7	1.2	65.2
	Algorithm 4	158.9	0.7	73.8	Algorithm 6	158.6	0.7	68.9
	Algorithm 5	157.9	0.8	73.8	Algorithm 7	158.6	0.7	68.9
4	Algorithm 1	133.7	0.3	82.3	Algorithm 1	133.7	0.3	78.2
	Algorithm 2	170.0	0.1	82.3	Algorithm 2	170.0	0.1	80
	Algorithm 3	134.7	0.4	81.7	Algorithm 3	134.7	0.4	77.3
	Algorithm 4	134.8	0.3	82.3	Algorithm 6	134.8	0.3	80
	Algorithm 5	134.8	0.3	82.3	Algorithm 7	134.8	0.3	80
5	Algorithm 1	147.6	2.6	66.2	Algorithm 1	147.6	2.6	60
	Algorithm 2	164.4	2.1	68.6	Algorithm 2	164.4	2.1	62.3
	Algorithm 3	146.0	1.2	75.1	Algorithm 3	146	1.2	69.1
	Algorithm 4	148.0	1.0	76.9	Algorithm 6	147.9	1.0	71.3
	Algorithm 5	147.9	1.0	76.9	Algorithm 7	147.9	0.9	71.3



(a) (b)

Figure 8. Estimated Playout Delay based on Algorithm 4 and Algorithm 5 (a) trace3 (b) trace1



(a) (b)

(c) (d)

Figure 9. Playout delay estimation for Algortihm 3 and Algorithm 6: (a) Algorithm 3 for Trace 1 (b) Algorithm 6 for Trace 1 (c) Algorithm 3 for Trace 2

(d) Algorithm 6 for Trace 2