Interactivity-Aware Playout Adaptation

Jochen Issing\textsuperscript{1}, Nikolaus Färber\textsuperscript{2}, and Reinhard German \textsuperscript{1}

\textsuperscript{1}Computer Networks and Communication Systems, Dept. of Computer Science, University of Erlangen, Germany
\textsuperscript{2}Fraunhofer Institut Integrierte Schaltungen, Erlangen

\texttt{jochen.issing@cs.fau.de, nikolaus.faerber@iis.fraunhofer.de, reinhard.german@fau.de}

Abstract

Adaptive Playout is a solution in IP-based communication clients to compensate network issues using a dynamic receiver buffer. Whereas small buffers provoke loss artefacts due to delayed packets, large buffers result in high delay and loss of interactivity. Existing Voice over IP clients balance the trade-off between low delay and loss artefacts based on empirical values. Most often the degree of interactivity is not taken into account and adaptation parameters remain the same for long monologues and lively discussions. We present a novel playout adaptation scheme that not only adapts to changing network conditions but also to conversational interactivity. Using an integrated approach we improve conversations with high interactivity significantly while at the same time preserve high audio quality. Using a full client implementation we verify our findings by conducting conversation tests and obtain gains of up to 0.9 Mean Opinion Score.

Index Terms: voice over ip, adaptive playout, conversational interactivity

1. Related Work

While most of the existing playout adaptive communication systems adapt to changing network conditions, e.g., jitter and loss, there are few existing studies trying to solve the trade-off between loss of interactivity and jitter compensation.

An early study \cite{1} proposed a perceptually motivated optimality criterion that allows the receiver to automatically balance packet delay versus packet loss. The dejitter buffer size is adaptively set and the adopted criterion relies on the use of a simplified version proposed by Cole and Rosenbluth \cite{2} of the conversational-quality E-Model. The dejitter buffer algorithm was taken from \cite{3}. The overall improvement with the adaptive approach is given to be about 14 R points, which corresponds to a Mean Opinion Score (MOS) score difference of about 0.75. However, actual conversation test results are missing.

Another study \cite{4} proposes a method for predicting voice quality for buffer design and optimization. In the method, nonlinear regression models are derived for a variety of codecs (e.g. G.723.1/G.729/AMR/LBC) with the aid of ITU PESQ \cite{5} and the E-model. The authors claim with preliminary results, that the proposed algorithm can achieve optimal perceived voice quality compared to other algorithms under all network conditions considered. The improvements are given on the MOS scale and are in the range of about 0.2 to 0.8. Conversation test results are, however, missing.

A paper by Sat et al. \cite{6} presents adaptive playout scheduling and loss concealment schemes for delivering consistent conversational voice communication quality perceived by users in real-time VoIP systems. The authors evaluate their adaptive playout scheduling and redundancy-based loss concealment schemes by packet traces collected in the PlanetLab. The authors design the algorithm based on conversation test results, but the system verification test is again accomplished based on objective criteria.

In summary, there are several interesting approaches towards the adaptation to both better jitter compensation and preserving conversational interactivity. However, all methods are based on the E-Model, which does not take the actual interactivity into account. Apart from that, none of the systems has been proven out scheduling and redundancy-based loss concealment schemes for real-time VoIP systems. The authors evaluate their adaptive playout algorithms for delivering consistent conversational interactivity. Our model consists of two main components:

- Model for the impact of late loss
- Model for the impact of delay with respect to interactivity.

Each model predicts the quality, based on the parameters late loss rate, end-to-end delay, or conversational interactivity. The compound model then adds up both predictions and searches for a local maximum to optimize the trade-off between reduced late loss and preserved conversational interactivity. Our model consists of two main components:

\hspace{1cm}• Model for the impact of late loss
\hspace{1cm}• Model for the impact of delay with respect to interactivity.

We do not include a model for the impact of time shrinking artifacts, as with high-profile time shrinking mechanisms like Waveform Similarity and Overlap Add (WSOLA), this impact is negligible \cite{7}.

2.1. Conversational Quality as a Function of Late Loss

The basis for our model is given by the results of previously conducted burst loss or concealment tests \cite{8}. We first build a
model from the results using non-linear least squares modelling using GNU/R [9] with the command nls. The non-linear model is given by the function prototype

\[ CQ_c = \frac{b}{a + r_c}, \]

with the two constants \( b \) and \( a \) and \( r_c \), representing the concealment rate. All coefficients show high significance due to low alpha error probability and thereby supporting the significance of the model. Good mode coverage is shown by an R^2 value of 0.7807 and an adjusted R^2 value of 0.7799.

### 2.2. Conversational Quality as a Function of Delay and Interactivity

The model to predict Conversational Quality (CQ) as a function of delay and interactivity is built based on the results from [10] where we decide on Speaker Alternation Rate (SAR) as our metric for interactivity. SAR is the number of speaker alternations per minute and a commonly used term in the related work [11] [12] [13]. We build a linearized model using GNU/R [9] with the command nls. The full model equation of the impact of late loss to CQ is given by:

\[ CQ_I = c + f_1 \cdot r_{SA} \cdot d^2 + f_2 \cdot d \cdot r_{SA}, \]  

(1)

with the constant \( c \), factors \( f_1 \) and \( f_2 \). End-to-end delay is denoted as \( d \) and SAR is given as \( r_{SA} \). The model yields an F-statistic p-value of \( < 2 \times 10^{-6} \) as well as an R^2 of 0.255 and an adjusted R^2 of 0.253. The low R^2 values are typical for the impact of delay on CQ and indicate missing model parameters based on the complexity of human conversation [14].

### 2.3. Combined Model

We now combine the two models from the previous two sections. For the combination, we apply additive combination of the artifacts, as also done in the E-Model [15]. The full model is given by

\[ CQ = CQ_0 - I_c - I_I \]  

(2)

with \( CQ_0 \) being the maximum achievable conversational quality (i.e., MOS 5), \( I_c \) representing the impact of late loss on CQ and \( I_I \) representing the impact of delay and SAR, our metric for interactivity.

The impact of late loss is calculated from the late loss model using

\[ I_c = CQ_0 - CQ_c \]  

(3)

with \( CQ_c \) being the CQ for 0 late loss and the impact of delay and interactivity using

\[ I_I = CQ_0 - CQ_I \]  

(4)

with \( CQ_I \) representing the CQ for no delay or zero SAR.

To find the solution for the trade-off between both models, an optimization algorithm maximizes the full model CQ in (2). Due to the fact that we can shift \( CQ_0 \) to the value of \( CQ_0 \) by setting parameter \( c \) without breaking the model’s estimation result, we can alternatively maximize (2) by instead maximizing

\[ \hat{CQ} = CQ_0 + f_1 \cdot r_{SA} \cdot d^2 + f_2 \cdot d \cdot r_{SA} + \frac{b}{a + r_c}, \]  

(5)

with \( f_1 \) and \( f_2 \) being the coefficients of the SAR and delay model, \( r_{SA} \) as SAR and \( d \) as delay, \( b \) and \( a \) being the coefficients of the late loss or concealment model and \( r_c \) representing the late loss rate.

### 3. Design of the Communication System

To verify the compound model and therefore the adaptation algorithm, we conduct conversation tests using a customized CS with the compound model integrated as the control for the jitter buffer. We have chosen the ACE implementation from Fraunhofer IIS Erlangen, as the basis for our CS. It is a low delay audio communication framework, which includes AAC-ELD, socket I/O and audio playout among other functionality. The ACE implementation is written purely in C++ and so are all of our extensions to it [16] [17] [8].

To make the system adaptive towards jitter, delay and CI, we have to extend its existing structure to collect the additionally required information and provide the necessary control structures. Figure 1 shows an overview of a Fraunhofer ACE client after we have extended it.

![Figure 1: Overview of the ACE CS client with additional adaptation](image)

Right before the decoder, the Buffer Control decides whether the packet needs to be decoded or discarded due to buffer shrinking. This module was already available before we extended the CS. However, it had to undergo major changes to integrate the full adaptation model for our studies. The buffer control contains the full optimization logic, transforms the jitter calculation which has been carried out with an arbitrary offset before to reflect actual absolute delay values and take our metric for interactivity into account.

To simulate different network conditions we integrated a network simulator module into the CS. For the implementation we loosely followed the implementation of G.1050 [18], which concatenates up to 20 servers for the core network simulation. To model our network, we have chosen to implement three different software modules that model delay, jitter, and delay spikes.

We need to have good estimates of absolute end-to-end delay for our model of CQ with respect to delay and interactivity. For this purpose, we send, receive, and analyze RTCP messages, which are sent on the same ports as the RTP packets in our implementation, as specified in [19]. As stated in [20], RTCP receiver reports may be sent at most every 5 s. To gain stable end-to-end delay estimation we increase the interval to 25 ms on both sender and receiver side.

As we conducted the verification test in our laboratory with easily controllable conditions, we decided to implement a simple level based silence detection. We are using a level
of $-44.626$ dBFS as our threshold for silence. Audio frames exceeding the threshold are classified as audible, audio frames below the threshold are classified dropable and thus can be used to reduce the jitter buffer and perform Adaptive Playout (AP).

Since the ITU general purpose Voice Activity Detection (VAD) as standardized in G.720.1 [21], did not provide stable results for high interactivity due to long hangover, we define a Level Activity Detection (LAD) algorithm using a band pass filter to suppress background noise and an activity threshold of $-35$ dBFS.

As the SAR is our input metric to adapt the playout towards CI, delay and late loss, we need to continuously estimate the SAR during the conversation in real time based on the time distance between two speaker alternations. The SAR estimation is implemented to quickly adapt to all changes in SAR.

Using the delay of each packet and the current SAR, the buffer control estimates the $CQ_I$ using the $CQ_I$ model as in (1) per each packet delay. To estimate the competing $CQ_c$, curve, the buffer control uses the sorted delay values from the jitter estimation.

Figure 2 shows the results of both models $CQ_I$ and $CQ_c$ as well as the combined model according to (5). The end-to-end delay at which $CQ$ is maximized corresponds to the optimal trade-off between delay and is taken as the target buffering time in the jitter buffer.

![Figure 2: Effect of packet delay and interactivity on estimated MOS. Results are shown using the same delay values for two different SAR values 5 and 20. The lower curves represent the individual models for $CQ_c$ and $CQ_I$. The sum of both models ($CQ$) for which the maxima are marked with a red dot is shown in the upper half of the plots. The corresponding delay value of each maxima is indicated using a dashed vertical line.](image)

5. Verification Test Results

70 subjects were tested in 35 groups, of which 8 test subjects had to be filtered out due to technical or operational problems. Figure 3 shows the histogram of the average SAR per single test with two test subjects.

![Figure 3: Histogram of SAR separated by test class. The range of SAR values has been quantized into 15 breaks, which are represented by the bins in the histogram. The different test classes are separated by color and add up to the overall value.](image)

The shortest average talk spur therefore was 900 ms (at 67/min), while the overall average talk spur lasts 3119 ms. The test comprised relative lively discussions on average.

Figure 4 shows the responses to overall quality (i.e., R4) on average, for the baseline and for the interactive CS depending on average SAR of the conversation test.

![Figure 4: Responses to overall quality depending on SAR over all tests. The graphs show the frequency of responses indicated by darkness, with dark regions representing high frequency. The results are shown on average for both CSs as well as for the baseline and interactive CS separately. Besides that, the linear regression line is shown in white to indicate the intensity of the dependency of SAR on MOS. Most notably is the lack of very high interactivity with the baseline CS. High end-to-end delay restricted the test subjects below a maximum SAR of 40/min. Besides that, the baseline CS shows stronger dependency of SAR on MOS than the interactive CS with MOS dropping from 4 to 2, while the MOS for the interactive CS drops from 4 to 3 only.](image)

After each test, we asked the test subjects about the quality of each CS and which they would prefer.

4. Design of the Verification Tests

To examine the effect of the full model on $CQ$, we exposed the test subjects to two different CS: the baseline system, which follows the network delay given a certain accepted late loss rate only (given by the white boxes in Figure 1) and the interactive system, which observes the CI continuously and adapts the buffer size appropriately (all boxes from Figure 1).

We have chosen to use the random number verification (RNV) test from [22]. As natural conversation, we use Richard’s task (Richard) from [22] where one test subject shall guess geometric shapes. As a no interactivity task, the test subjects listen to a short sequence of an audio book (Book).
Figure 5 shows the voting results for the preferred CS of all test subjects, as given in the questionnaire after every second conversation test.

![Vote Percentage Chart](chart.png)

Figure 5: Test subject rating for the preferred CS with 95% confidence intervals as given by the t-test. The results are separated by test class (i.e., rnv, richard, and book) alongside with the overall average. The height of each bar comprises the number of test subjects voting for the baseline CS, the interactive CS or where the subjects could not decide between the two.

The results for the rnv class are very clearly in favor of the interactive CS. More than 80% have preferred the interactive to the baseline CS and about 10% of the test subjects could not decide between both.

For medium interactivity like with richard’s task, the baseline CS was preferred by about 50% of all users, while approximately 25% have preferred the interactive CS. Even though the confidence intervals overlap slightly, this results discovers missing pieces in the optimization model.

For the book task, the test subjects had highest votes for the interactive CS with a rate of roughly 50% but looking at the confidence intervals we can not prove statistical significance. On average, almost 50% of all test subjects preferred the interactive CS to the baseline CS, for which about 30% have voted and less than 20% were unsure about their decision. In summary we can conclude that the interactive CS has been proven to improve the conversations of the conversation tests on average, but also has room for improvements in the case of medium interactivity.

Figure 6 shows the expected MOS value for each test class (task) as well as on average. The interactive CS yields at least the same quality of experience as the baseline CS except for richard’s task. While the MOS of the baseline CS drops down to 2.2, the interactive CS provides an MOS of about 3.0.

For richard’s task, the adaptation was too aggressive and should be adjusted to work more moderately as discussed previously. The confidence intervals of all tests are widely constant.

### 6. Conclusion

We have build a novel integrated audio Communication System for playout adaptation. The CS adapts the audio playout to balance the impact of delay on Conversational Quality and artifacts caused by late loss. The adaptation is performed depending on the current Conversational Interactivity to adjust the amount of adaptation to the characteristics of the ongoing conversation.

We enhanced an existing CS with a network simulation, round-trip-time estimation, voice activity detection, speaker alternation detection, and buffer control to measure, assess and apply the PA accordingly. We exposed 70 test subjects to both the baseline and the interactive CS and assessed the impact of the CS type statistically.

On average, almost 50% of all test subjects preferred the interactive CS to the baseline CS, for which about 30% have voted and less than 20% were unsure about their decision. The average test task has improved from a MOS score of 3.3 to a MOS score of 3.5, for medium interactivity the MOS score did suffer slightly by 0.2 MOS points, which indicates the necessity of further refinements of the optimization model. For very high interactivity tasks the MOS score improved from 2.2 to 3.1, which is a significant improvement.

In summary we can conclude that the interactive CS has been proven to improve the conversations of the conversation tests on average, but also has room for improvements in the case of medium interactivity.

### 7. References


