Towards Statistical Analysis of the Impact of Playout Buffer on Quality of Experience in VoIP Applications

Tibor Gyires Yongning Tang Aishwarya Mishra Olusegun Obafemi
School of Information Technology
Illinois State University
Normal IL 61790 USA
tbgyires,ytang,amishra,oeobafe@ilstu.edu

Abstract—**The speech quality of Voice over IP (VoIP) applications can be assessed subjectively as Quality of Experience (QoE) and objectively as Quality of Service (QoS). QoE is multifaceted, which ties together user perception and expectations to application, network performance, and various voice data processing (e.g., codec) and streaming (playout buffering) methods. Most of prior work focuses on understanding the impact of network performance on QoE, but not explicitly describing how playout buffer affects user satisfaction or QoE assessment. Towards this goal, this paper presents a statistical analysis of the correlation among QoE assessment, QoS measurement, and the impact of playout buffer on QoE assessment. In this paper, we first identify QoE as a function along two dimensions of network loss and delay to understand how different network factors as well as playout buffer affect QoE assessment. Then, we propose a new performance metric called playout buffer QoE impact factor (IF^{QoE}) to explicitly evaluate the effectiveness of playout buffer in terms of its contribution to QoE improvement. Finally, we validate IF^{QoE} to statistically show its accuracy in terms of its strong correlation with the results of QoE assessment. All our study is based on extensive simulations using various emulated or real network scenarios. Our simulation results show that IF^{QoE} can accurately evaluate the impact of playout buffer on QoE assessment using directly measurable network performance metrics.**

Keywords- Quality of Service, Quality of Experience, Playout Buffer, Impact Factor, Statistical Analysis, VoIP.

I. Introduction

In recent years, *Voice over IP* (VoIP) along with other multimedia networking applications has become one of the most important IP network services to end users. Correspondingly, a major paradigm shift on the quality assessment methods of multimedia networking applications has occurred from network-centric to user-centric. User perceived *Quality of Experience* (QoE) is given special attention by network operators and service providers to assess the overall level of users satisfaction and maintain acceptable quality of service for VoIP communications.

User perceived QoE in VoIP is generally described in terms of *Mean Opinion Score* (MOS) [4], the formal subjective measure of user satisfaction on received voice quality. QoE is multifaceted, which ties together user perception and expectations to application, network performance, and various voice processing (e.g., codec) and streaming (e.g., playout buffering) methods.

Most of prior work focuses on understanding the impact of network performance on QoE, but not explicitly describing how playout buffer affects user satisfaction or QoE assessment.

A typical VoIP application buffers incoming packets and delays their playout in order to compensate for variable network delays (i.e., jitter). Such an application buffer is commonly referred to as *Playout Buffer*. A playout buffer can allow late arrival packets to be smoothly played out. However, the fluctuating end-to-end network delays may enforce the size of a playout buffer to increase to a level to trigger user unsatisfactory delay. In addition, if the size of playout buffer is too small, some late arrival packets will still be dropped in playout buffer because their arrival time exceeds required presentation deadlines. The two conflicting goals of minimizing buffering delay and minimizing late packet loss have motivated various playout algorithms.

Our objective is to understand the impact of playout buffer on QoE in VoIP applications. In the paper, we study the correlations among network delay, network loss, buffering delay, buffering loss, and QoE. Our study aims at providing an easy-to-measure performance metric to accurately evaluate the effectiveness of a playout buffer on improving QoE assessment.

In this paper, we use a simple but representative evaluation model to study the correlation among QoE assessment, QoS measurement, and the effectiveness of playout buffer in terms of its contribution to QoE improvement. In this process, we first identify QoE as a function along two dimensions of network loss and delay to understand how different network factors as well as playout buffer affect QoE assessment. Then, we propose a new performance metric called playout buffer QoE impact factor (IF^{QoE}) to explicitly evaluate the effectiveness of playout buffer in terms of its contribution to QoE improvement. Finally, we validate IF^{QoE} to statistically show its accuracy in terms of its strong correlation with the results of QoE assessment. Our extensive simulations show that IF^{QoE} can accurately evaluate the impact of playout buffer on QoE assessment using measurable network performance metrics.

Our contribution is twofold: (1) we present an experimental study on measuring the dimensions of QoE assessment, and (2) we propose a new playout buffer performance metric called playout buffer QoE impact factor (IF^{QoE}), and provide a statistical analysis on the validation and accuracy of IF^{QoE} on evaluating the impact of playout buffer on QoE.

Though various approaches on showing QoS-QoE correlation have been proposed in the literature as described in Section II, to the best of our knowledge, none of them focuses on explicitly describing the impact of playout buffer on QoE assessment. After reviewing two basic QoE assessment methods in Section III, we elaborate our analytical methodology and propose IF^{QoE} in Section IV. We continue our study by first showing QoE dimensioning results in Section V, and then present a statistical analysis on the validation of IF^{QoE} in Section VI. Finally, we conclude our work in Section VII.

II. Related Work

There are numerous approaches proposed to objectively measure speech quality in VoIP. Robinson and Yedwab [10], [25] proposed a Voice Performance Management system to monitor call quality in real-time by proactively monitoring, alerting, troubleshooting and reporting network performance problems. Robinson and Yedwab [10] concluded that only packet loss, jitter and latency show the correlations between QoS and QoE.

Gierlich and Kettler [13] provided insight into the impact of different network conditions and the acoustical environment on speech quality. Testing techniques for evaluating speech quality under different conversational aspects were also described. Gierlich and Kettler [13] argued that there is no single number that can objectively indicate speech quality; and pointed out that overall speech quality is a combination of different single values from different speech quality parameters. Wang et. al., [14] designed and implemented a QoS-provisioning system that can be seamlessly integrated into current Cisco VoIP systems. Wang et. al., [14] also described Call Admission Control (CAC) mechanisms (Site-Utilization-based CAC and Link-Utilization-based CAC) to prevent packet loss and over-queuing in VoIP systems.

Myakotnykh and Thompson [15] described an algorithm for adaptive speech quality management in VoIP communications, which can show a real-time change in speech encoding parameters by varying voice packet sizes or compression (encoding) schemes. The algorithm involves the receiver making control decisions based on computational instantaneous quality level (which is calculated per talkspurt using the E-Model) and perceptual metric (which estimates the integral speech quality based on latency, packet loss and the position of quality degradation period in the call). Myakotnykh and Thompson [15] calculated the maximum achievable quality level for a given codec under specific network conditions, packet playout time, packet delay before jitter buffer and degradation in quality caused by traffic burstiness and high network utilization. The algorithm however results in an increase in average quality without increasing individual call quality.

Raja, Azad and Flanagan [16] designed generalized models to predict degradation in speech quality with high accuracy, in which genetic programming is used to perform symbolic regressions to determine Narrow-Band (NB) and Wide-Band (WB) equipment impairment factors for a mixed NB/WB context. Zha and Chan [17] described two algorithms for objective measurement of speech quality: single-ended (needing only to input the degraded speech signal) and double-ended (needing both the original and degraded speech signals). The algorithm developed by Zha and Chan [17] can objectively measure in real-time speech quality using statistical data mining methods.

Several algorithms have also been proposed to optimize some of the existing ITU-T models. The goal of optimization is to enhance existing models by correcting weaknesses that are identified in the models. Gardner, Frost and Petr [18] proposed an algorithm to optimize the E-Model by considering coder selection, packet loss, and link utilization. The authors however stated that the algorithm would have to be enhanced if used in a wide area network involving multiple user session. Mazurczyk and Kotulski [19] proposed an audio watermarking method based on the E-Model and the MOS, which provides speech quality control by adjusting speech codec configuration, playout buffer size and amount of Forward Error Correction (FEC) mechanism in VoIP under varying network conditions.

One of the limitations of the E-model is the fact that the model does not consider the dynamic nature of underlying networks that support VoIP. This limitation is addressed by several authors designing adaptive playout buffering to improve voice quality in VoIP. Most of these studies either optimize the E-Model, the PESQ [5] or combine the PESQ and the E-Model to propose a more holistic solution. Mazurczyk and Kotulski [19] highlighted two problems that are associated with adaptive playout buffering: how to estimate current network status and how to transfer network status data to the sending or receiving side. Wu et. al. [20] admitted that VoIP playout buffer size has long been a challenging optimization problem, as buffer size must balance the dynamics of conversational interactivity and VoIP speech quality. They stated that the optimal playout buffer size yields the highest satisfaction in a VoIP call. They further investigated the playout buffering dimensions in Skype, Google Talk and MSN Messenger, and concluded that MSN Messenger produces the best performance in terms of adaptive playout buffering, while Skype does not adjust its playout buffering at all.

$$MOS = \begin{cases} 1, & if\ \mathcal{R} \leq 0 \\ 4.5, & if\ \mathcal{R} \geq 100 \\ 7 \times 10^{-6}\mathcal{R}(\mathcal{R}-60)(100-\mathcal{R}) + 0.035\mathcal{R} + 1, & otherwise \end{cases} \tag{1}$$

Narbutt and Davis [21] stated that the management of playout buffering is not regulated by any standard and is therefore vendor specific. They proposed a scheme that extends the E-Model and provides a direct link to perceived speech quality, and evaluated various playout algorithms in order to estimate user satisfaction from time varying transmission impairments including delay, echo, packet loss and encoding scheme.

III. Quality of Experience Assessment

In this section we discuss two commonly used and well accepted quality of experience assessment methods: mean opinion score (MOS) and E-Model.

A. Mean Opinion Score

Mean Opinion Score or MOS has been endorsed by ITU-T as a subjective method to evaluate voice transmission quality. The MOS test involves using a group of testers (listeners) to assign a rating to a voice call. The quality is rated on a scale of 1 to 5, with $1 = bad$, $2 = poor$, $3 = fair$, $4 = good$ and $5 = excellent$ [2]. The arithmetic mean of the scores provided by all listeners becomes the final MOS value of the voice call. Assessment ratings can also be obtained by clustering the test results as "Good or Better" or as "Poor or Worse", and further calculating the relative ratio or percentage of each type of results. For a given voice call, these results are expressed as "Percentage Good or Better" (%GoB) and "Percentage Poor or Worse" (%PoW) [3]. Table I shows the MOS rating, %GoB, %PoW and the correlation between each rating [4].

Table I: Subjective Ratings for Measuring QoE

User Satisfaction	**MOS** (5)	**%GoB** (100)	**%PoW** (0)
Very Satisfied	4.3-4.4	97.0-98.4	0.2-0.1
Satisfied	4.0-4.29	89.5-96.9	1.4-0.19
Some Dissatisfied	3.6-3.9	73.6-89.5	5.9-1.39
Many Dissatisfied	3.1-3.59	50.1-73.59	17.4-5.89
Nearly All Dissatisfied	2.6-3.09	26.59-50.1	37.7-17.39
Not Recommended	1.0-2.59	0-26.59	99.8-37.69

The advantage of the MOS is that it can provide an off-line analysis of end-user opinions. However, MOS tests cannot provide an absolute reference for the evaluations; that is, MOS ratings are dependent on the expertise of listeners [1]. Furthermore, MOS tests cannot be used in large scale experiments that involve a large number of users because of the involved overhead (e.g., test setup). Moreover, MOS tests are unrepeatable by nature.

B. E-Model

The E-Model, standardized by the ITU in 1998 as Recommendation $G.107$, provides a method for calculating a single metric representing voice quality, referred to as the *R-factor*, which can then be converted to estimate MOS values as shown in Eq. 1.

The E-Model is designed to measure the instant user perceived quality instead of the cumulative effect during an entire conversation. The E-Model assumes that individual impairment factors are additive on a psychological scale and combines the cumulative effects of these factors into the R-factor. The R-rating is on a scale of 0 to 100, with high values of R between 90 and 100 interpreted as excellent quality, while lower values of R indicate a lower quality. Values of R below 50 are considered unacceptable and values above 94.15 are assumed to be unobtainable in narrowband telephony. The E-Model measures individual impairment factors at different points in time to compute the R-rating. The value of the R-rating is consequently associated with measurements taken at a given time point and does not reflect the dynamic nature of quality during the entire length of a conversation.

The R-factor is expressed as the sum of five terms:

$$\mathcal{R} = \mathcal{R}_0 - I_s - I_d - I_e + A \tag{2}$$

R_O represents the basic signal-to-noise ratio, including noise sources such as circuit noise and room noise. The factor I_s is a combination of all impairments which occur simultaneously with the voice signal. The factor I_d represents the impairments caused by delay, and the effective equipment impairment factor I_e represents impairments caused by low bit-rate codecs and packet-losses of random distribution. The advantage factor A corresponds to the user allowance due to the convenience when using a given technology.

The E-Model not only takes in account the transmission statistics (transport delay and network packet loss), but it also considers the voice application characteristics, like the codec quality, codec robustness against packet loss and the late packets discard. However, the impact of playout buffer is simply converted into the impact of buffering delay and buffering loss, and thus not explicitly represented in E-Model.

In this paper, we are interested in finding the correlations of network performance (delay and packet loss) and user satisfaction assessment (MOS), and further relate these factors to the impact of playout buffer on QoE. Thus, we will adopt the recommended default values by the ITU-T

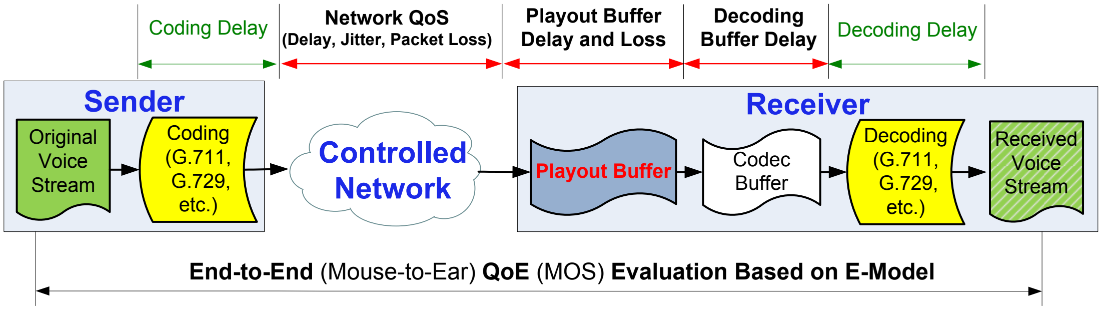


Figure 1: The Design of VoIP Speech Quality Assessment in Controlled Network Experiments.

Rec. G.107 for those intangible quantities (i.e., $\mathcal{R}_0$, A, I_s) and reduce the expression for the R-factor to:

$$\mathcal{R} = 94.2 - I_d - I_e \tag{3}$$

In the context of this work, delay impairment I_d comes from three sources: codec delay, network delay and playout buffering delay; and loss impairment I_e results from network packet loss and playout buffering packet loss.

IV. Analytical Methodology

In this section, we elaborate the network model and analytical methodology used in our study.

A. The Network Model

We generalize a typical VoIP application as a network system depicted in Fig. 1, which consists of a sender (caller), a receiver (callee), and a fully controlled network. On the sender, a voice stream is digitalized via a coding process, and then packetized to voice packets to send out. On the receiver, the received voice packets are first buffered in a playout buffer to compensate for network delay variation (jitter), and then further buffered in a codec buffer required by a decoding process.

It is worth noting that the playout and codec buffers are completely different from both their design objectives and their impacts on QoE assessment. A playout buffer is designed to allow the incoming voice packets with variant intervals (due to network jitter) can be played out as smooth as possible. Thus, a fixed or varying playout buffer delay is unavoidable depending on different buffering modes (fixed or adaptive); and moreover, some incoming voice packets may be dropped by the playout buffer if their arrival time later than required presentation deadlines. On the other side, codec buffer is required by decoding algorithm such that a minimum number of voice packets can be accumulated necessary for a decoding process being conducted. A codec buffer will cause a fixed buffering delay, but no packet loss.

Our study is performed in a well-known credible network simulation platform OPNET [11], which allows us (1) to choose a variety of codec schemes, (2) to create realistic networks supporting measurable performance metrics, (3) to flexibly control playout buffer; and (4) to estimate MOS (the result of QoE assessment) using E-Model.

In our study model, the sender can continuously send voice stream using a selected codec to the receiver over the network. The network can be fully controlled with specified network delay and loss rate to simulate various network conditions. A fully configurable playout buffer is presented on the receiver, which can operate in either fixed or adaptive mode with different parameters, including maximum buffer size, resizing interval, sliding mean coefficient.

According to Eq. 3, the impact of various components (the network, codec components, and playout buffer) on QoE assessment results from the total end-to-end accumulated delay (d_{tot}) and packet loss (e_{tot}). Since we consider the impact of coding and decoding delays into I_s, $d_{tot} = d_{net} + d_{buff} + d_{cbuff}$, and $e_{tot} = e_{net} + e_{buff}$. Here, d_{net}, d_{buff} and d_{cbuff} are delays caused by the network, playout buffer, and codec buffer, respectively. e_{net} and e_{buff} are packet loss rates caused by the network and playout buffer, respectively.

We proceed our experimental study in the following steps:

- To detect *Minimum Codec Buffer* (MCB): We remove the playout buffer and set the network to an ideal condition with a minimum constant network delay and no packet loss. We then gradually increase the size of codec buffer from the lowest value ($1ms$) to a more than enough large value (e.g., $250ms$), and use the measured MOS values of a continuous voice steam from the sender to the receiver to analyze the required minimum codec buffer for a specific codec, which will be further discussed later. Apparently, in such an ideal network condition, the playout buffer is unnecessary (no delay variation). Therefore, once the codec buffer reaches the corresponding MCB for a given codec, the measured MOS value should present a clear jump when the codec buffer size is changed from right below MCB to MCB.
- To investigate QoE dimensions using network loss and delay: We still keep the playout buffer removed, and control the network with various constant delays and loss rates. With all network conditions, we use the

measured MOS values of a continuous voice steam to find the user tolerable QoE boundary dimensioned in network loss and delay.

- To validate the new proposed playout buffer QoE impact factor IF^{QoE}: We validate the accuracy of IF^{QoE} for measuring the effectiveness of a playout buffer on QoE improvement. Specifically, for a given network condition, we configure two VoIP systems with and without playout buffer, respectively. We use the measured MOS values in these two cases to evaluate the improvement of QoE, which is compared to the results according to the computation of IF^{QoE}. We present a statistical study to show the accuracy of IF^{QoE} in measuring the impact of playout buffer on QoE. Finally, we use IF^{QoE} to evaluate several playout strategies in VoIP applications.

B. Experimental Design

For simplicity of presentation, we show in Table II all configurable parameters of our study model and the measured objects.

Table II: Configurable Parameters and Measurable Results

Configuration Parameters & Their Settings	
Codec	Encode/decode schemes (G.711, etc.)
Network Discard Ratio	The percentage of packets dropped
Network Latency	Delay dist, fixed values, scripted dist
Buffer Sizing Interval	Playout Buffer Resizing time
Maximum Buffer Size	Measured by buffer delay
Sliding Mean Coefficient	Coefficient for new talkspurt data
Playout Mode	fixed or adaptive buffer size

Measured Objects & Their Implications	
MOS	Estimated mean opinion score
Jitter	delay variation
Instant Playout Buffer Delay	the same as current buffer size in ms
Instant Playout Buffer Loss	pkt loss rate due to large pkt intervals
Network Loss Rate	ratio of lost pkts in network
End-to-end Delay	the total pkt delay from mouth to ear
Traffic Sent	Average received pkts/bytes per second
Traffic Received	Average sent pkts/bytes per second

For each experiment run with a specific setting, we keep the sender continuously sending voice stream to the receiver for one hour, and take 100 samples every second for all measured objects. We repeat 100 runs for each experiment and report the corresponding sample means. Please note, the actual execution time for each run is much shorter than the simulated running period. For example, the average execution time for a one-hour run is only $36s$ in a regular PC with Intel Core 2 Duo 2.66 GHz CPU and 3 GB memory.

C. Playout Strategies

Most of the adaptive playout algorithms described in the literature perform continuous estimation of the network delay and its variation to dynamically adjust the talkspurt playout time. Standard adaptive playout algorithm is based on Jacobsons work on TCP round trip time estimation. The algorithm estimates two statistics: the delay itself and its variance as shown in Eq. 4 and Eq. 5, and uses them to calculate the playout time [12].

$$\widehat{d}_i = (1-\alpha) \times \widehat{d}_{i-1} + \alpha \times n_i \tag{4}$$

$$\widehat{v}_i = (1-\alpha) \times \widehat{v}_{i-1} + \alpha \times |\widehat{d}_i - n_i| \tag{5}$$

Here, $\widehat{d}_i$ is the estimated amount of time from when the i^{th} packet is generated by the sender until it is played out at the receiver; n_i is the total delay introduced by the network. $\widehat{v}_i$ is the delay variance of i^{th} packet. α is called sliding mean coefficient in our study ($0 \leq \alpha \leq 1$).

Several other methods were also introduced to better estimate network delay. For example, instead of using a single sliding mean coefficient, two different sliding mean coefficients were used to adapt more quickly to short burst of packets incurring long delays. The idea behind the different playout strategies described in this paper is simple and all follow the so-called absolute timing method as defined by Montgomery [23].

If both the propagation delay and the distribution of the variable component of network delay are known, a fixed playout delay can be computed such that no more than a given fraction of arriving packets are lost due to late arrival. In such approach, the playout delay is fixed either for the length of the voice call, or is recalculated at the beginning of each talkspurt.

One potential problem with this approach is that the propagation delay is not known (although it can be estimated and typically remains fixed throughout the duration of the voice call). A more serious concern is that the end-to-end delay distribution of packets within a talkspurt is not known, and can change over relatively short time scales.

An approach to dealing with the unknown nature of the delay distribution is to estimate these delays and adaptively respond to their change by dynamically adjusting the playout delay. In this study, we define four playout strategies to describe such delay estimation and dynamic playout delay adaptation. As we will see, these strategies determine a playout delay on a per-talkspurt basis. Within a talkspurt, packets are played out in a periodic manner, thus reproducing their periodic generation at the sender. However, the playout strategies may change the playout delay from one talkspurt to the next, and thus the silence periods between two talkspurts at the receiver may be artificially elongated or compressed (with respect to the original length of the corresponding silence period at the sender). Compression or expansion of silence by a small amount is not noticeable in the played out speech.

When playout buffer resizing is necessary, an appropriate new buffer size can only be estimated, which also reflects the estimation of the network condition before next resizing opportunity. Algorithm 1 shows a commonly used dichotomic search algorithm for computing new buffer size. In this algorithm, first, an expected MOS value is calculated with new buffer size set to the average of maximum and minimum buffer sizes (line 2). Then, the new (expected) MOS value is used to update the smaller one between the MOS values when choosing the minimum and maximum buffer sizes, respectively (line 3-9). Finally, the algorithm chooses the buffer size that generates a higher MOS value (line 10-13). It is worth noting that the buffer size is not proportional to the MOS value, and thus it is possible that MOS$_{min}$ may be larger than MOS$_{max}$ (line 10).

Algorithm 1 PlayoutBufferResizing()

```
1:  while (BuffSize$_{max}$− BuffSize$_{min}$ $> 1$) do
2:      MOS$_{current}$ $\leftarrow$ MOSCompute((BuffSize$_{max}$+ BuffSize$_{min}$)$/2$)
3:      if MOS$_{min}$ $<$ MOS$_{max}$ then
4:          MOS$_{min}$ $\leftarrow$ MOS$_{current}$
5:          BuffSize$_{min}$ $\leftarrow$ (BuffSize$_{max}$+ BuffSize$_{min}$)$/2$
6:      else
7:          MOS$_{max}$ $\leftarrow$ MOS$_{current}$
8:          BuffSize$_{max}$ $\leftarrow$ (BuffSize$_{max}$+ BuffSize$_{min}$)$/2$
9:      end if
10:     if MOS$_{min}$ $>$ MOS$_{max}$ then
11:         BuffSize $\leftarrow$ BuffSize$_{min}$
12:     else
13:         BuffSize $\leftarrow$ BuffSize$_{max}$
14:     end if
15: end while
16: return  BuffSize
```

Clearly, these control parameters discussed above play important role in the performance of a playout buffer in terms of its impact on QoE assessment. In this paper, we denote a playout strategy s as a tuple: <Buffer Sizing Interval τ, Sliding Mean Coefficient α, Maximum Buffer Value ν >, or simply $<\tau, \alpha, \nu>$. Buffer Sizing Interval τ decides how often the adaptive resizing should be decided. For example, resizing can be taken at the moment between talkspurts or in a fixed periodic interval (e.g., $10ms$). Sliding Mean Coefficient α is a coefficient for new spurt data to compute the playout buffer size, which can be set empirically. For example, as the experimental results shown in [12], α was set to 0.998002 in a single parameter estimation function as Eq. 4, or two different values in a double parameter estimation function with $\alpha = 0.998002$ for increasing trends in the delay and $\alpha = 0.75$ for decreasing trends. Maximum Buffer Value ν specifies the maximum buffer limit, which is measured in the delay experienced by a packet in the buffer.

D. Playout Buffer QoE Impact Factor: IF^{QoE}

Essentially, a playout buffer is designed to improve QoE, especially when experiencing fluctuating network delays. To the best of our knowledge, there is no prior work showing how to practically and accurately evaluate the effectiveness of a playout buffer from the perspective of QoE improvement. In this section, we tackle this challenge by proposing a new performance metric for playout buffer evaluation.

Recalling our discussion in Section III-B, we have presented the R-factor as the following function, which has been also shown previously in Eq. 3 with $I_d = \mathcal{F}(d_{tot})$ and $I_e = \mathcal{G}(e_{tot})$:

$$\mathcal{R} = 94.2 - \mathcal{F}(d_{tot}) - \mathcal{G}(e_{tot}) \tag{6}$$

Both $\mathcal{F}()$ and $\mathcal{G}()$ are monotonically increasing functions. Assuming that the same voice stream is sent over the same network to two VoIP systems with the only difference that one has playout buffer (denoted as S_{buff}) and another one does not (denoted as S_{nobuff}). The playout buffer in S_{buff} will introduce buffering delay and buffering loss, which does not appear in S_{nobuff}. With the above assumption, we have the following conclusion:

$$\begin{aligned} \mathcal{R}_{buff} &= 94.2 - \mathcal{F}(d_{net} + d_{buff}) - \mathcal{G}(e_{net} + e_{buff}) \\ \mathcal{R}_{nobuff} &= 94.2 - \mathcal{F}(d_{net}) - \mathcal{G}(e_{net}) \end{aligned} \tag{7}$$

The above equations imply that $\mathcal{R}_{buff} \leq \mathcal{R}_{nobuff}$ is always true, which apparently contradicts our intuition. The contradiction results from mistakenly calculated $\mathcal{G}(e_{net})$ in $\mathcal{R}_{nobuff}$. For a network with varying delays, the received VoIP packets may be dropped due to their varying arrival intervals that cannot meet their presentation deadlines required by the decoding process on the receiver. We refer to such packet loss due to missing playout buffer as e_{nobuff}. Thus, we rewrite the above equation Eq. 7as:

$$\begin{aligned} \mathcal{R}_{buff} &= 94.2 - \mathcal{F}(d_{net} + d_{buff}) - \mathcal{G}(e_{net} + e_{buff}) \\ \mathcal{R}_{nobuff} &= 94.2 - \mathcal{F}(d_{net}) - \mathcal{G}(e_{net} + e_{nobuff}) \end{aligned} \tag{8}$$

In order to make $\mathcal{R}_{buff} > \mathcal{R}_{nobuff}$, the following condition should hold:

$$\mathcal{G}(e_{net}+e_{nobuff})-\mathcal{G}(e_{net}+e_{buff}) > \mathcal{F}(d_{net}+d_{buff})-\mathcal{F}(d_{net}) \tag{9}$$

The condition above clearly shows the tradeoff between two conflicting design objectives of playout buffer to minimize both d_{buff} and e_{buff}. A good playout algorithm should pay minimal cost d_{buff} to gain maximum reward $e_{nobuff} - e_{buff}$. To fairly evaluate different playout strategies in terms of QoE improvement, the new performance metric of playout buffer should indicate both the absolute QoE gain (denoted as Q_{gain}) and the relative QoE gain ratio (denoted as Q_{ratio}) as defined in Eq. 10:

Considering various empirical functions proposed for practically calculating $\mathcal{F}(d_{tot})$ and $\mathcal{G}(e_{tot})$ (e.g., [24]), the relation between $\mathcal{F}(d_{tot})$ and d_{tot} can be regressed to a linear function; and a logarithmic line can fit the correlation curve between $\mathcal{G}(e_{tot})$ and e_{tot}. According, we propose IF^{QoE} as the new performance metric for playout buffer shown in Eq. 11.

$$\begin{aligned} Q_{gain} &= Q_{buff} - Q_{nobuff} = [\mathcal{G}(e_{net}) - \mathcal{G}(e_{net} + e_{buff})] - [\mathcal{F}(d_{net} + d_{buff}) - \mathcal{F}(d_{net})] \\ Q_{ratio} &= \frac{Q_{buff} - Q_{nobuff}}{Q_{nobuff}} = \frac{[\mathcal{G}(e_{net}) - \mathcal{G}(e_{net} + e_{buff})] - [\mathcal{F}(d_{net} + d_{buff}) - \mathcal{F}(d_{net})]}{94.2 - \mathcal{F}(d_{net}) - \mathcal{G}(e_{net} + e_{nobuff})} \end{aligned} \tag{10}$$

$$IF^{QoE} = \quad [\mathcal{G}(e_{nobuff}) - \mathcal{G}(e_{buff})] \times \frac{\mathcal{F}(d_{nobuff})}{\mathcal{F}(d_{buff})} \tag{11}$$

Intuitively, the more the reward indicated by$\mathcal{G}(e_{nobuff}) - \mathcal{G}(e_{buff})$ and the less the cost indicated by $\frac{\mathcal{F}(d_{nobuff})}{\mathcal{F}(d_{buff})}$, the higher the IF^{QoE}.

To analyze the accuracy of IF^{QoE}, we adopt the following two commonly used empirical functions introduced in [24]. Here, the empirical function for $\mathcal{G}(e)$ is specific to G.711. Similar functions exist for other codecs, but will not be discussed in this paper.

$$\begin{aligned} \mathcal{F}(d) &= \quad 0.024d + 0.11(d - 177.3)H(d - 177.3) \\ \mathcal{G}(e)_{G.711} &= \quad 30\ln(1 + 15e)H(0.04 - e) + \\ & \quad\ 19\ln(1 + 70e)H(e - 0.04) \end{aligned} \tag{12}$$

where $H(x)$ is the Heavyside (or step) function such that:

$$H(x) = \begin{cases} 0, & if\ x < 0 \\ 1, & if\ x \geq 0 \end{cases} \tag{13}$$

In the case of packet loss rate greater than 4%, which is used in our following study, we can calculate IF^{QoE} as the following:

$$IF^{QoE} = \quad 19\frac{d_{nobuff}}{d_{buff}} \times \ln\frac{1 + 70 \times e_{nobuff}}{1 + 70 \times e_{buff}} \tag{14}$$

Among the four parameters in Eq. 14, e_{buff} and d_{buff} are commonly obtained by monitoring the impact of playout buffer on packet loss and delay. d_{nobuff} can be calculated using Eq. 4 to estimate end-to-end delay between the sender and receiver. Different codec has different jitter tolerance. For example, G.711 can tolerate jitter up to $20ms$. For obtaining e_{nobuff}, we first use the information from RTP header to estimate the current network jitter. Then all incoming voice packets with jitter more than the tolerance will be counted as dropped ones to estimate e_{nobuff}.

V. QoE Dimensioning

In this section, we first identify minimum codec buffer. Then we present our study on QoE dimensioning using network loss and delay.

A. Minimum Codec Buffer

With respect to voice over IP, a codec is an algorithm used to encode and decode the voice conversation. A original analog voice signal needs to be converted (or encoded) to a digital format suitable for transmission over the Internet. Once at the other end, it needs to be decoded for the receiver. There are a variety of Codecs available and many of which utilize compression in order to reduce the required bandwidth of the conversation. The impairment of Codec on QoE comes from two aspects: (1) compression reduces the signal to noise ratio, and (2) when heavy compression is used, it takes time which adds a delay to conversation.

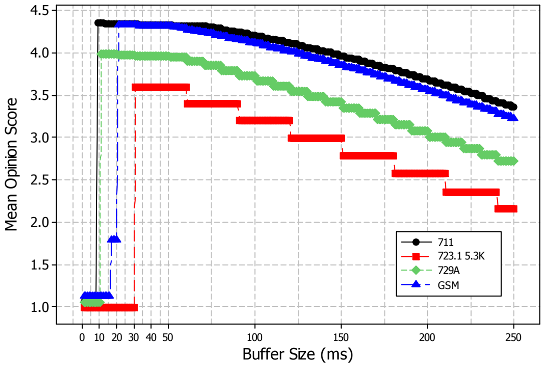


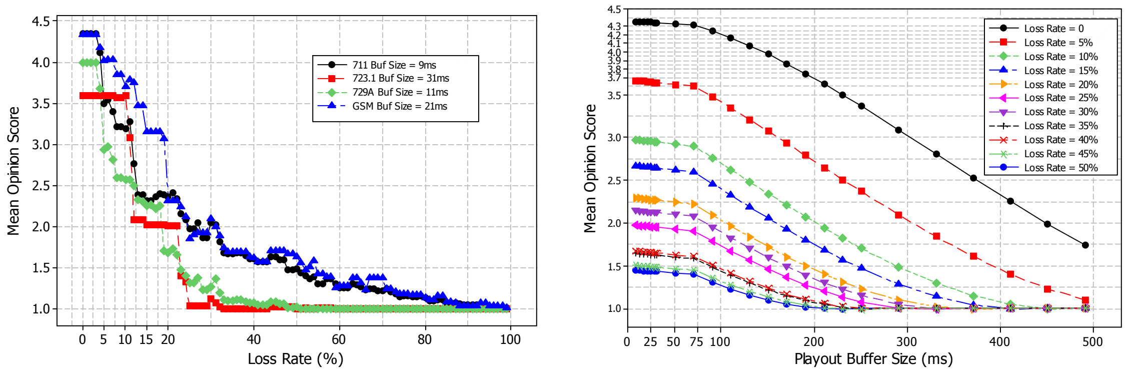
Figure 2: The Impact of Minimum Codec Buffer on MOS.

To experimentally find the MCB for each codec, we set the network to an ideal condition with only a minimum constant network delay and no network loss. Then we increase the codec buffer size from $1ms$ to $250ms$. In such an ideal network condition, a close to maximal MOS is expected if the codec buffer size is set to MCB. Thus, we use the measured MOS with increasing codec buffer size to detect the MCB for each codec. Fig. 2 shows the experiments results with clearly detected MCB. However, when the codec buffer size is further increasing after MCB, the MOS value decreases due to the extra delay incurred at the expanding codec buffer.

Table III: Minimum Codec Buffer and MOS

CODEC	$< BelowMCB, MOS >$	$< MCB, MOS >$
G.711	$< 8, 1.06 >$	$< 9, 4.35 >$
G.723.1	$< 30, 0.99 >$	$< 31, 3.59 >$
G.729A	$< 10, 1.05 >$	$< 11, 3.98 >$
GSM	$< 20, 1.79 >$	$< 21, 4.33 >$

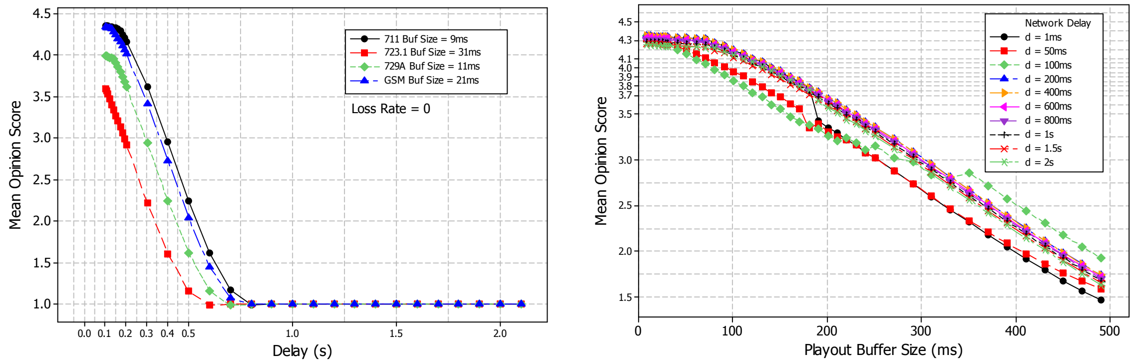
We summarize the Minimum Codec Buffer (MCB) of four investigated codec and their corresponding MOS values in an ideal network condition in Table III. The second column shows when codec buffer cannot reach MCB (only $1ms$



(a) The Impact of Loss on MOS with Minimum Playout Buffer Size.

(b) The Impact of Constant Loss Rate on MOS with Varying Playout Buffer Size (codec = G.711).

Figure 3: The Impact of Loss



(a) The Impact of Delay on MOS with Minimum Playout Buffer Size.

(b) The Impact of Constant Delay on MOS with Varying Playout Buffer Size (codec = G.711).

Figure 4: The Impact of Constant Delay

less), the corresponding MOS value is significantly low (e.g., 1.06 for G.711). In contrast as shown in the third column, when the codec buffer size is set to MCB (e.g., $9ms$ for G.711), the MOS value reaches its maximum (e.g., 4.35 for G.711) when the network is in an ideal condition. In our study, we use time delay to measure buffer size.

B. The Impact of Network Loss

Network loss can significantly degrade user satisfaction on received VoIP data. We conducted a variety of experiments and use the measured MOS values to find the user tolerable boundary impacted by various network losses. In these experiments, we choose four codecs: G.711, G.723.1, G.729A and GSM with their codec buffer sizes set to their specific MCB as in Table III. We control the network loss rates varying from 0% to 100%.

Fig. 3(a) depicts how network loss could seriously degrade user satisfaction in a VoIP application no matter which codec is used. For example, for GSM codec, when the network loss rate increases to 15% or beyond, most users cannot tolerate the perceived voice quality, which is indicated by the boundary MOS value 3.5. Similarly, the user tolerable boundaries for network loss when using G.711, G.723.1 and G.729A are 9%, 13% and 7%, respectively.

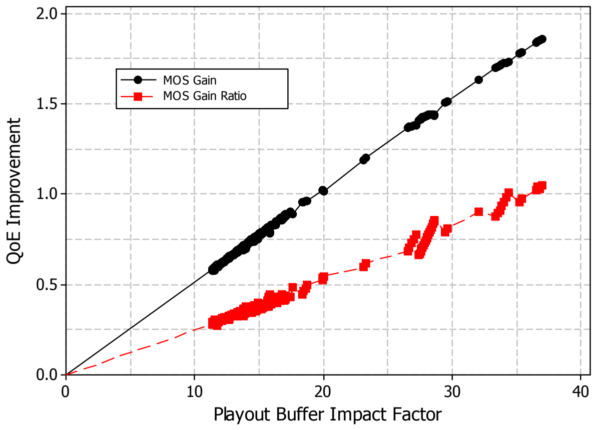
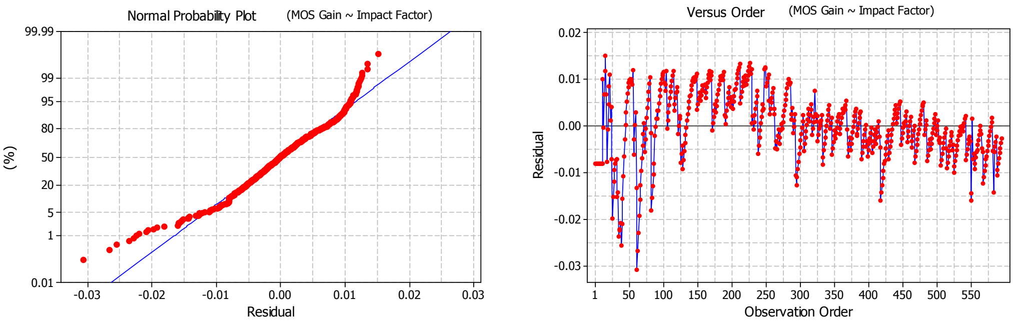


Figure 5: Validity of IF^{QoE}.

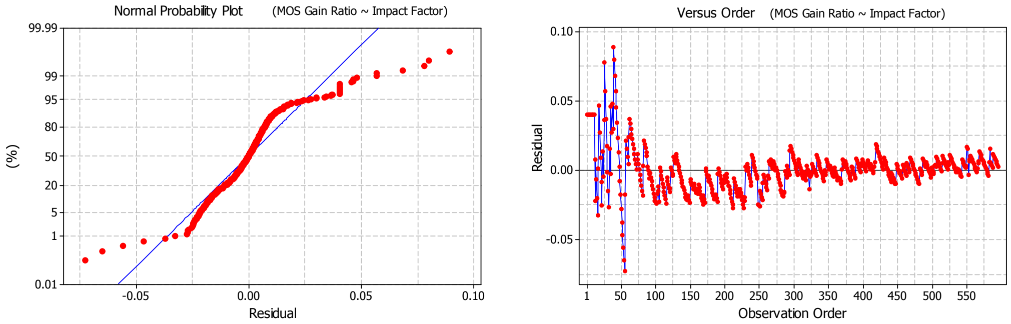
We further verify if a VoIP application with playout



(a) Normal Probability Plot of MOS Gain.

(b) Residual versus Order of MOS Gain.

Figure 6: Residual Analysis for MOS Gain $\sim IF^{QoE}$



(a) Normal Probability Plot of MOS Gain Ratio.

(b) Residual versus Order of MOS Gain Ratio.

Figure 7: Residual Analysis for MOS Gain Ratio $\sim IF^{QoE}$

buffer can have any positive impact on degraded user satisfaction due to network loss. For this purpose, we control the network loss rate increased from 0% up to 50%, and vary playout buffer size from $0ms$ to $500ms$. We use the measured MOS values to analyze the impact of playout buffer. The experiment results are shown in Fig. 3(b), which clearly confirms that playout buffer cannot improve the user satisfaction on received voice quality impaired by network loss, and even worse, it may further degrade user satisfaction due to the unnecessary playout buffer delay.

C. The Impact of Constant Network Delay

In this section, we continue our study on measuring QoE in another dimension: network delay. Similarly, we conducted experiments and use the measured MOS values to analyze the user satisfaction tolerable boundary impacted by different constant network delays. In these experiments, network loss rate is set to 0. We choose the same codecs with their codec buffer sizes set to their specific MCB. We vary network delays from $0ms$ to $2{,}000ms$.

Fig. 4(a) depicts how constant network delays could seriously degrade user satisfaction in a VoIP application for all selected codecs. For example, for G.711, when the constant network delay increases up to $350ms$ or more, most users cannot tolerate the perceived voice quality, which is again indicated by the MOS value 3.5. Similarly, the user satisfaction tolerable boundaries due to different constant network delays when using G.723.1, G.729A and GSM are $100ms$, $250ms$ and $300ms$, respectively.

We also verify if a playout buffer can help in such situation. For this purpose, we set network delay in each experiment to a constant value, and increase it from $1ms$ up to $2{,}000ms$, and vary playout buffer size from $0ms$ to $500ms$. The experiment results are shown in Fig. 4(b), which clearly confirms that playout buffer cannot improve the user satisfaction impaired by constant network delays, and even worse as the previous case, it may further degrade user satisfaction due to unnecessary playout buffer delay.

VI. IF^{QoE} Validation

In this section, we validate and analyze the accuracy of IF^{QoE} in evaluating the effectiveness on improving QoE of a playout buffer.

We conducted similar experiments as we discussed in Section IV-D. In these experiments, the sender sends the same voice stream over the network with controlled delay distribution to two VoIP systems. The only difference between these two systems is that one has playout buffer (denoted as S_{buff}) and another one does not (denoted as S_{nobuff}). For each sampled value in each experiment run, we use the measured MOS values from both S_{nobuff} and S_{buff} to calculate MOS gain and MOS gain ratio.

Meanwhile, we derive the corresponding IF^{QoE} using $e_{nobuff}, e_{buff}, d_{nobuff}$ and d_{buff}.

The result is reported in Fig. 5, which indicates a strong linear correlation between IF^{QoE} and MOS_{gain}, as well as between IF^{QoE} and MOS_{ratio}. In order to be more specific, we denote QoE_{gain} and QoE_{ratio} as MOS_{gain} and MOS_{ratio}.

Simple linear regression shows us the following two linear correlation functions:

$$MOS_{gain} = 0.00800 + 0.0507 \times IF^{QoE} \tag{15}$$

$$MOS_{ratio} = -0.0403 + 0.0282 \times IF^{QoE} \tag{16}$$

The coefficients of determination or r^2 for the two linear regression functions $MOS_{gain}(IF^{QoE})$ and $MOS_{ratio}(IF^{QoE})$ are 99.9% and 98.8%, respectively, which clearly shows that IF^{QoE} is a valid performance metric in measuring the effectiveness on QoE improvement of playout buffer.

A. Residual Analysis

To illustrate the accuracy of IF^{QoE}, we show residual plots for both $MOS_{gain}(IF^{QoE})$ and $MOS_{ratio}(IF^{QoE})$ in Fig. 6 and Fig. 7, respectively. In both Fig. 6(a) and Fig. 7(a), the residuals close to zero and as moving farther away from zero fewer residuals appear, which prove that the condition of normality is clearly met for both regression functions. The randomness shown in Fig. 6(b) and Fig. 7(b) further confirms the fitness of the regression functions.

VII. Conclusion

By identifying QoE as a function along two dimensions of network loss and delay, we have shown how different network factors as well as playout buffer can affect QoE assessment. Then, we have proposed a new performance metric called Playout Buffer QoE Impact Factor or IF^{QoE} for evaluating the effectiveness of playout buffer on QoE improvement. IF^{QoE} can be calculated using directly measurable performance metrics, which can accurately represent the effectiveness of a playout buffer on both absolute and relative QoE improvement. IF^{QoE} is the first proposed method bridging QoE assessment, QoS measurement and the evaluation on the impact of playout buffer. Our future work will include applying IF^{QoE} to evaluate specific playout algorithms used in real wired and wireless (e.g., WiFi and WiMax) network environments.

References

[1] Sat, B., and Wah, B. W. (2009). Analyzing Voice Quality in Popular VoIP Applications. IEEE MultiMedia, vol. 16, no. 1, pp. 46-59.

[2] Zwar, E. J., and Munch, B. (2006). Voice Quality and Network Capacity Planning for VoIP.

[3] Narbutt, M., and Davis, M. (2005). Assessing the Quality of VoIP Transmission Affected by Playout Buffer Scheme. 4th International Conference on Measurement of Speech and Audio Quality, Prague, Czech Republic.

[4] International Telecommunications Union (1996). ITU-T P.800. Methods for Subjective Determination of Transmission Quality.

[5] International Telecommunications Union (2007). ITU-T P.862 Corrigendum. Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs.

[6] Telecommunications Industry Association (2005). Telecommunications, IP Telephony Equipment and Voice Quality Recommendations for IP Telephony.

[7] Morris, M. G., Venkatesh, V., Davis, G. B., and Davis, F.D. (2003). User Acceptance of Information Technology: Toward a Unified View.

[8] Becvar, Z., Mach. P., and Bestak, R. (2009). Impact of Handover on VoIP Speech Quality in WiMAX Networks. Eighth International Conference on Networks, icn, pp.281-286, Gosier, Guadeloupe, France.

[9] ITU-T Recommendation G.107 (1998). The E-Model, a computational model for use in transmission planning.

[10] Robinson, P. and Yedwab, D. (2009). Voice and Video Application Performance Management in UC Deployments.

[11] OPNET Technologies: http://www.opnet.com/. Last accessed: 2/20/2012.

[12] R. Ramjee, J. Kurose, D. Towsley, H. Schulzrinne (1994) Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks. In Proceeding of IEEE INFOCOM.

[13] Gierlich, H. W. and Kettler, F. (2006). Advanced speech quality testing of modern telecommunication equipment: an overview. Signal Processing, 86(6), 1327 - 1340.

[14] Wang, S., Mai, Z., Xuan, D., and Zhao, W. (2006). Design and Implementation of QoS-Provisioning System for Voice over IP. IEEE Transactions on Parallel and Distributed Systems, vol. 17, no. 3, pp. 276-288.

[15] Myakotnykh, E. S. and Thompson, R. A. (2009). Adaptive Speech Quality Management in Voice-over-IP Communications. Fifth Advanced International Conference on Telecommunications, aict, pp.64-71, Venice/Mestre, Italy.

[16] Raja, A., Azad, R. M. A., and Flanagan, C. (2008). VoIP Speech Quality Estimation in a Mixed Context with Genetic Programming. 10th Annual Conference on Genetic and evolutionary computation, Atlanta, Georgia, United States.

[17] Zha, W. and Chan, W. (2005). Objective Speech Quality Measurement Using Statistical Data Mining. EURASIP Journal on Applied Signal Processing, no. 9, 1410-1424.

[18] Gardner, M., Frost, V.S. and Petr, D.W. (2003). Using optimization to achieve efficient quality of service in Voice over IP networks. IEEE International Performance, Computing, and Communications Conference, Phoenix, Arizona, United States.

[19] Mazurczyk, W. and Kotulski, Z. (2007). Adaptive VoIP with Audio Watermarking for Improved Call Quality and Security. Journal of Information Assurance and Security 2, 226-234.

[20] Wu, C., Chen, K., Huang, C., and Lei, C. (2009). An Empirical Evaluation of VoIP Playout Buffer Dimensioning in Skype, Google Talk and MSN Messenger. Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital and Video, Williamsburg, VA, United States.

[21] Narbutt, M. and Davis, M. (2005). Assessing the Quality of VoIP Transmission Affected by Playout Buffer Scheme.

4th International Conference on Measurement of Speech and Audio Quality, Prague, Czech Republic.

[22] Mohamed, S., Cervantes-Perez, F., and Afifi, H. Integrating Network Measurements and Speech Quality Subjective Scores for Control Purposes. IEEE Infocom, Anchorage, Alaska, (2001)

[23] W. Montgomery. Techniques for Packet Voice Synchronization. IEEE Journal on Selected Areas in Communications, Sol. SAC-6, No. 1 (Dec. 1983), pp. 1022 - 1028.

[24] Cole, R.G. and J. Rosenbluth, Voice Over IP Performance Monitoring, Journal of Computer Communications Review, vol. 4, no. 3, April (2001).

[25] O. Obafemi, T. Gyires and Y. Tang. An Analytic and Experimental Study on the Impact of Jitter Playout Buffer on the E-model in VoIP Quality Measurement. The 10th International Conference on Networks, 2011