

Online Estimation of VVoIP Quality-of-Experience via Network Emulation

Haytham Assem[†], Mohamed Adel^{*}, Brendan Jennings^{*}, David Malone[†],
Jonathan Dunne⁺ and Pat O’Sullivan⁺

[†]*Hamilton Institute, National University of Ireland Maynooth, Ireland*

^{*}*TSSG, Waterford Institute of Technology, Ireland*

⁺*IBM Software Lab, Dublin, Ireland*

E-mail: [†]{hitham.salama.2012, david.malone}@nuim.ie

^{*}madel@tssg.org, bjennings@ieee.org

⁺{jonathan.dunne, patosullivan}@ie.ibm.com

Abstract — We describe a testing framework that can provide online estimates of audio and video call quality on network paths, without requiring either end-user involvement or prior availability of audio/video sequences or network traces. The framework includes a tool that emulates the audio and video traffic of IP calls and employs an extended E-Model to measure the audio quality and VQM to estimate video quality. Additionally, it can emulate network impairments to run experiments in different network conditions. Our experiment results show that the quality measurements acquired using the framework compare well to the most commonly applied industry standard for objective voice and video offline testing—PESQ and PSNR respectively.

Keywords — VVoIP, QoE, PESQ, E-model.

I INTRODUCTION

Communications traffic is increasingly being carried over IP networks, with Voice and Video over IP (VVoIP) applications becoming popular in both the public Internet and in enterprise networks. Since IP networks do not guarantee end-to-end delay, packet loss rates, jitter and available bandwidth, monitoring and estimating the VVoIP call quality in prevailing network conditions is essential to mitigating issues that can significantly reduce the Quality-of-Experience (QoE) as experienced by end users.

In a VVoIP application, voice/video is digitized and packetized at the sender before its transmission over the IP network to the receiver. At the receiver the packets are decoded and played out to the listener as shown in Figure 1. The process of converting an analog voice signal to digital is done by an audio/video “codec.” Codecs vary in bandwidth required, latency, sample period, frame size and the maximum achieved end user perceived quality, thus different codecs are better suited to

different network conditions.

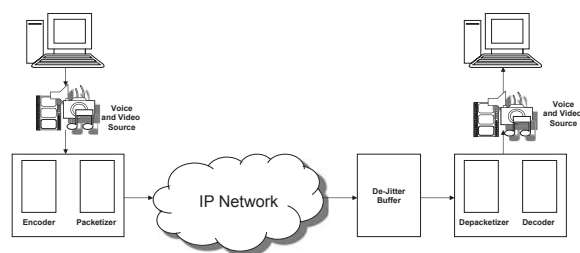


Fig. 1: High-level VVoIP System Architecture.

ITU-T outlines two test methods to assess QoE: *subjective testing* and *objective testing*. Subjective testing was the earliest approach to evaluating the quality by assigning Mean Opinion Scores (MOS). ITU-T Rec. P.800 [1] presents the MOS subjective test procedures for audio quality testing. It usually involves 12-24 participants who individually listen to an audio stream for several seconds and rate the audio quality on the scale of 1 (Poor) to 5 (Excellent). Similarly, BT.500 [2] presents

a methodology to obtain MOS values for video quality. Subjective testing using MOS is time consuming, expensive and does not allow for real time measurements. Several techniques were developed for measuring MOS in an objective way (without human perception): PESQ [3, 4] and E-model assess audio quality, while PSNR (Peak Signal to Noise Ratio) and Video Quality Model (VQM), as outlined in ITU-T Rec. G.1070 [5] are used for obtaining video quality.

For audio, PESQ automatically maps its score to a subjective MOS score. It takes into account two signals: one is the reference signal while the other one is the actual degraded signal. Both the signals are processed by a tool that uses the PESQ algorithm and outputs a MOS score. The major drawbacks of PESQ approach are that it does not take into account impairments such as acoustic echo, transmission delay and that it can not be used for real time monitoring purposes. In contrast, the E-model technique, specified in ITU-T Rec. G.107 [6], is a non-intrusive method that uses network metrics locally monitored at the sender to estimate call quality, so it can be used for live call monitoring. One drawback with the E-model is that it requires knowledge of a so-called “impairment factor” of the codec, which ITU-T provide for codecs they specify, but which are not specified for a range of other commonly used codecs.

For video, PSNR works in a similar way to PESQ. This method assesses the performance of video transmission systems by calculating PSNR between the original and the received (degraded) video; it is a differential metric which is computed using images. On the other hand, VQM provides an online technique to calculate video quality based on various factors including current network conditions, codec used and properties of the transmitted sequence. In most cases, transmission of video can be subjected to a lot of losses. Moreover, delay can cause unwanted pauses in the received signal, as the receiver might need to pause it processing, while the buffer refills. Consequently, both packet loss and delay will cause degradation in the interactive video call quality between the end users.

Since processing audio/video sequences is time consuming and computationally intensive, existing objective techniques are not ideal for online VVoIP QoE and since audio/video codecs have different characteristics and usually it is impossible to define in advance the most appropriate codec to use. Given this, we focus in this paper on the use of a novel testing framework to estimate the voice/video call quality in advance. Such estimates can then be used to select the most appropriate codec to use for upcoming calls. Crucially, this processes does not require the transfer of source

audio/video sequences and does not require the end user to provide quality rankings.

The paper is organized as follows: in §II we place our work in the context of the recent published literature on the topic. In §III, we provide a brief description of the used measuring audio/video call quality metrics. In §IV, we propose the implementation of our framework. The results of our framework compared to other standard industry metrics are shown in §V. §VI conclude the paper.

II RELATED WORK

Real time Voice and Video over IP applications are sensitive to network conditions—variations in metrics including end-to-end delay, packet loss rates and jitter have a significant impact on the quality as perceived by end users. Given this, monitoring and estimating call quality is an important task that has been extensively studied by the research community. We briefly review some of the relevant previous work on this topic.

Jiang et al. [7] introduce a voice quality monitoring system based on the SIP protocol, which uses RTP statistics to get MOS score using the simplified E-model. Kim et al. [8] propose a network performance monitoring method that uses RTCP statistics to monitor multimedia services like VoIP and IPTV. da Silva et al. [9] analyze the QoS provided by SIP for voice traffic by measuring the delay, jitter and packet losses. Carvalho et al. [10] propose three corrections to the E-model in order to give more accurate results indicating the QoE expected at the end user; they also describe a measurement tool based on these corrections. Gong et al. [11] propose a pentagram model to measure the QoE based on service integrality, service retainability, service availability, service instantaneousness and service usability. Due to also the lack of QoE monitoring systems, Hershey et al. [12] propose a new approach that aggregates observations from real time applications running on net-centric enterprise systems. They show their results on several VoIP scenarios including a Denial-of-Service event that causes noticeable application delay. Calyam et al. [13] propose GAP-Model, which assess VVoIP QoE via an offline model of QoE that is expressed as a function of bandwidth, delay, jitter, and packet loss.

In contrast to the works describe above we present a QoE assessment framework that emulates the traffic for different audio/video codecs operating under different network conditions and uses measurements to estimate likely QoE without requiring end user input or the comparison of source and degraded sequences. The framework can be applied in an online manner by a VVoIP application for codec selection and admission control purposes given its measurements of prevailing

network conditions.

III MEASURING CALL QUALITY

In this section, we present the audio and video quality models used to estimate QoE in our framework. In particular, we describe how we estimated the impairment factor used in the audio quality model (E-model) for a number of non ITU-T codecs.

a) E-Model

The E-Model is an objective model proposed by ITU-T G.107 [6]. It takes into account various parameters factors that affect the speech quality and it calculates a ‘‘Rating Factor’’ R in the range 0 to 100, where $R = 0$ represents the worst quality and $R = 100$ represents the best quality. R is calculated as in Eq.1:

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

R_0 is the Signal to Noise ratio (S/N) at 0 dBR point, I_s represents the speech voice impairments, I_d is the impairments occurred due to the delay, I_e is the impairment due to the equipment (e.g.: codecs and packet loss) and A is the advantage factor (e.g.: $A = 0$ for wireline). As outlined in [14, 15, 16] the E-model can be utilized to be used in the speech quality evaluation over VoIP-Based Communication Systems and the R factor expression can be reduced as expressed as:

$$R = 93.2 - I_d(d) - I_{e-eff} \quad (2)$$

I_d is a function of the one way delay only and can be calculated by the approximated formula expressed in Eq.3 (from [15]):

$$I_d = 0.024 \times d + 0.11 \times (d - 177.3) \times H(d - 177.3) \quad (3)$$

$$\text{where } H(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

I_{e-eff} is function of the codec used and the packet loss rate, it can be expressed by:

$$I_{e-eff} = I_e + (95 - I_e) \frac{Ppl}{Ppl + Bpl} \quad (4)$$

Here I_e represents the impairment factor given by codec compression, Bpl represents the codec robustness against random losses and Ppl represents measured network packet loss rate. The values of I_e and Bpl are given only for ITU codecs in ITU-T G.113 appendix [17] as neither the impairment factors of all the codecs factors are provided nor

Table 1: Derived Linear Regression Model Parameters for Different Codecs.

Parameters	GSM	ILBC	SPEEX	SILK
a	22.931	20.836	28.244	18.3442
b	0.1555	0.762	0.2043	1.54894
c	42.175	18.013	27.423	1.31953

Table 2: Relationship between R and Mean Opinion Score.

R	Satisfaction Level	MOS
90-100	Very satisfied	4.3+
80-90	Satisfied	4.0-4.3
70-80	Some users dissatisfied	3.6-4.0
60-70	Many users dissatisfied	3.1-3.6
50-60	Nearly all users dissatisfied	2.6-3.1
0-50	Not recommended	1.0-2.6

can they be calculated easily. ITU-T recommendation G.113 does not provide codec I_e and Bpl values for the most commonly used codecs like ILBC, SILK, GSM and SPEEX. To establish these values we, for each of these codecs, estimate MOS using the PESQ method by directly comparing reference and degraded voice signals. In [18], we derived a non linear regression model for each codec by the least squares method and curve fitting. The derived I_{e-eff} model has the following form:

$$I_{e-eff} = a \log(1 + b \times Ppl) + c \quad (5)$$

The Ppl in (5) is the packet loss rate and the parameters (a , b and c) have the values shown in Table 1 for the different codecs.

Finally, the R is converted to Mean Opinion Score (MOS) using Eq.6. Table 2 shows the relationship between MOS and R .

$$MOS = \begin{cases} 1 & \text{if } R < 0 \\ 1 + 0.035R + \\ \quad R(R - 60)(100 - R) \times \\ \quad 7 \times 10^{-6} & \text{if } 0 \leq R \leq 100 \\ 4.5 & \text{if } R > 100 \end{cases} \quad (6)$$

b) Video Quality Model (VQM)

ITU-T G.1070 [5] specified a video quality model for telephony services. Video quality V_q is defined as:

$$V_q = 1 + I_{coding} \exp\left(\frac{-P_{plv}}{D_{Pplv}}\right) \quad (7)$$

V_q represent the MOS value ranging from 1 to 5. Coding losses due to combinations of video bit

rate (Br_v [kbit/s]) and video frame rate (Fr_v [fps]) is represented by I_{coding} . D_{Pplv} is the measure of robustness for the video quality against packet loss, where percentage of packet loss rate is defined by $Pplv$ [%]. I_{coding} and D_{Pplv} are further defined in [5] by the following set of equations:

$$I_{coding} = I_{Ofr} \exp\left(-\frac{(\ln(Fr_v) - \ln(Ofr))^2}{2D_{FrV}^2}\right) \quad (8)$$

Ofr is the optimal frame rate where video quality is the maximum. I_{Ofr} is the maximum quality at each video bit rate (Br_v). They are expressed as:

$$Ofr = v1 + v2 \times Br_v \quad (9)$$

$$I_{Ofr} = v3 - \frac{v3}{1 + \left(\frac{Br_v}{v4}\right)^{v5}} \quad (10)$$

D_{FrV} defines the robustness of video quality due to frame rate (Fr_v):

$$D_{FrV} = v6 + v7 \times Br_v \quad (11)$$

The degree of video quality robustness against packet loss is defined by D_{PplV} expressed in (12).

$$D_{PplV} = v10 + v11 \times \exp\left(-\frac{Fr_v}{v8}\right) + v12 \times \exp\left(-\frac{Br_v}{v9}\right) \quad (12)$$

Finally, coefficients $v1, v2...v12$ are defined according to codec type, key frame interval, video display size and video format.

IV FRAMEWORK IMPLEMENTATION

In the first part of this section, we introduce the components used in our framework while in the second section, we show how we fit all of the components together giving our approach in measuring the audio and video QoE.

a) Framework components

Our framework uses software called Iperf [19] to measure packet loss, jitter and throughput. Iperf is a networking tool that creates TCP and UDP data streams of specified size; it runs on various platforms including Linux, UNIX and Windows.

The voice and video packets are sent using UDP. Consequently, the exact measurement of the delay between the sender and destination is not directly measured. We use Ping to send Internet Control Message Protocol (ICMP) echo request packets to

the target destination and wait for the ICMP response. To get an accurate measurement of the delay we emulate the Ethernet-layer bandwidth according to the codec using Dummynet [20] and set the ping parameters based on the codec used.

We use Dummynet in our tool for two purposes. First, to change the network conditions (delay, packet loss, queue and bandwidth) to be able to test the QoS and QoE under different network conditions. Second, to set the bandwidth with the Ethernet bandwidth according to the codec emulated in order to measure accurate delay results with the current browsing sessions if any on the computer.

b) Development of the framework

Our framework measures the QoS of the network based on the codec used and maps it to a QoE MOS score indicating the end user satisfaction level expected during the call. The Packet size (Ps) and the Ethernet bandwidth (Eb) varies from codec to another. In our framework we calculated them as:

$$Ps = Fs \times framesPerPacket + ipHeader + eOverHead \quad (13)$$

$$Eb = Ps \times \left(\frac{bw}{Fs}\right)_{Codec} \quad (14)$$

Ps is the total packet size, Fs is the frame size according to the codec (see Table I), $framesPerPacket$ is the number of frames per packet, $ipHeader$ equals 40 bytes composed of the IP, UDP and RTP headers, $eOverHead$ equals 38 bytes composed of the preamble, Ethernet header, CRC and Ethernet Inter-Frame Gap, bw is the bandwidth required by the codec.

For video transmission, H.264 is not transmitted using fixed packet length, but the packet length changes dynamically according to the available bandwidth in order to attain an acceptable video quality and to minimize the effect of distortion. Roughly, for transmission of low quality video, 300 Kbit/s of available bandwidth is needed, whilst for high quality 500 Kbit/s would be required. HD video requires a minimum of 1.5 Mbit/s bandwidth to be available at both ends of the call. We investigated the variation of packets length under the previous bandwidths in an interval of 60 seconds then took the mean packet length in order to reach an approximation for the packet length at different bandwidths for emulating the video traffic; the results are in Table 3.

Before measuring the QoS of the network and the QoE expected at the end user, the network conditions can be emulated for testing the robustness

Table 3: Mean Packet Length Estimates for H.264.

Bandwidth <i>Kbit/s</i>	Mean packet length <i>Bytes</i>
300 (Low Quality)	316
500 (High Quality)	637
1500 (HD)	885

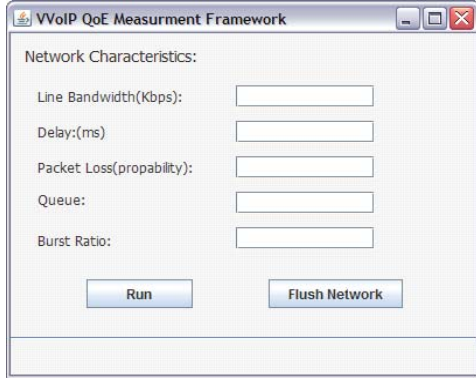


Fig. 2: Inputting Data for Network Emulation.

of different codecs under different network conditions. Fig 2 shows the dialogue box for inputting this data. The IP destination address, port number, codec used and frames/packet are the main inputs before running the testing framework; Dummynet will then emulate the network conditions. The delay is measured using Ping command taking in its account the packet size and the sending bit rate of the codec used as calculated in Eq.13,14. Iperf is called to measure the packet loss percentage, throughput and jitter by specifying Datagram size (Eq.13) and Ethernet Bandwidth (Eq.14) for audio, or by using Table 3 for video to create appropriate data stream according to the codec that will be used during the call. By measuring the throughput which is considered the performance ceiling, we are able to calculate the number of calls that a certain link can carry safely. We can state that a particular link will carry no more than X G.711 calls or Y G.729A calls or Z H264 calls:

$$nOfCalls = \lfloor \frac{throughput}{Eb} \rfloor \quad (15)$$

$nOfCalls$ is the number of calls that can be carried through a particular link safely, $throughput$ is the average rate of successful message delivery over a communication channel and Eb is the Ethernet bandwidth required according to the codec used. In order to increase the accuracy, average QoS network factors are measured by repeating the previous procedures 5 times and taking the average. At the end the QoS parameters measured are mapped to QoE MOS score using E-model and VQM described in section III. The pseudocode for

this process is shown in Algorithm 1.

```

input   : Destination IP, Destination Port No, Codec
            used, Video format, video frame rate and video
            bit rate (For Video testing only).
output  : QoS factors of the current network conditions,
            QoE MOS ranking and user satisfaction level.

begin
  Step1: Emulate Network.
  Initialize Dummynet emulator by loading kernel
  module;
  Emulate network conditions (Line Bandwidth, Delay,
  Random Packet loss, Burst Ratio, Queue length);
  Step2: Initialize Test.
  Check codec selected for call;
  Specify packet size, inter-packet time and sending bit
  rate;
  Step3: Begin Test.
  Counter = 0;
  while Counter less than 5 do
    Start Packet trains from source to destination;
    Measure one-way delay using ICMP request;
    Measure packet loss, throughput and jitter using
    Iperf;
    Increment Counter;
  end
  Calculate average results for one way delay, packet
  loss, throughput and jitter;
  Calculate link capability (No of Calls);
  Step4: Display Measured QoS factors.
  Display previous extracted data;
  Calculate QoE MOS score using E-model for audio
  and VQM for video;
  Display MOS score and user satisfaction level;
  Step5: End Test.
  Flush all inbound/outbound pipes of Dummynet;
end

```

Algorithm 1: QoE Estimation Process.

V RESULTS AND DISCUSSION

In this section, we provide the results of our QoE estimation process for voice and video in comparison to the most commonly applied industry standard for objective voice and video quality testing: PESQ and PSNR. In order to measure the accuracy of our results, we used a beta version of IBM Sametime Unified Telephony (SUT) [21] product, measuring the audio/video call quality under different packet loss rates using Dummynet. We have compared these to offline audio and video testing using PESQ and PSNR respectively. We first outline the results for audio and then outline the video results.

a) Audio Testing

A screenshot of the configuration dialog box for audio testing is shown in Figure 3. Tests are carried out on several codecs: G711, G723.1 5.3k, G723.1 6.4k, G726, G729, G729 A, GSM FR, SILK, ILBC and SPEEX. We show a sample of our results in Figure 4 and 5. The x-axis represents the packet loss rate ranges from 0-20% and the y-axis indicates the MOS from the framework and PESQ algorithm. Our results match well with the PESQ scores, confirming the accuracy of our approach. We observe in our results that we slightly underestimate MOS compared to scores produced from PESQ. This can be explained by the observation

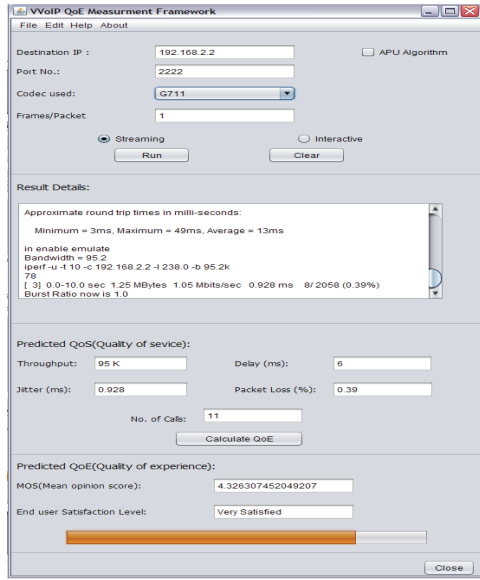


Fig. 3: Screenshot of Audio Testing GUI.

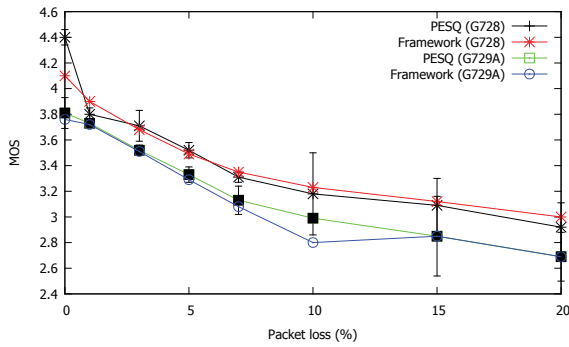


Fig. 4: MOS estimations for G.728 and G.729A audio codecs.

that we take into our account the delay impairment factor (conversational call quality) while the intrusive methods as PESQ do not take it into consideration.

b) Video Testing

Figure 6 shows a screenshot of our configuration dialog box for video testing. We compared our results to real time PSNR values of H.264 codec after converting them to MOS values. Table IV (derived by Ohm [22]) is used to map the PSNR to MOS values that can be used to estimate perceived quality. Our results match well the PESQ scores indicating the accuracy of our approach. We interpolate between the values in Table 4 by assuming that the relation between MOS and PSNR inside these regions is linear.

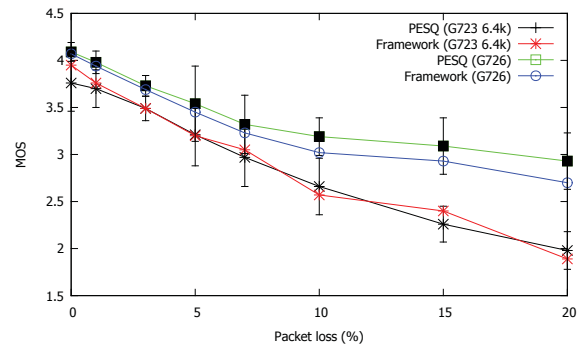


Fig. 5: MOS estimations for G.723 and G.726 audio codecs.

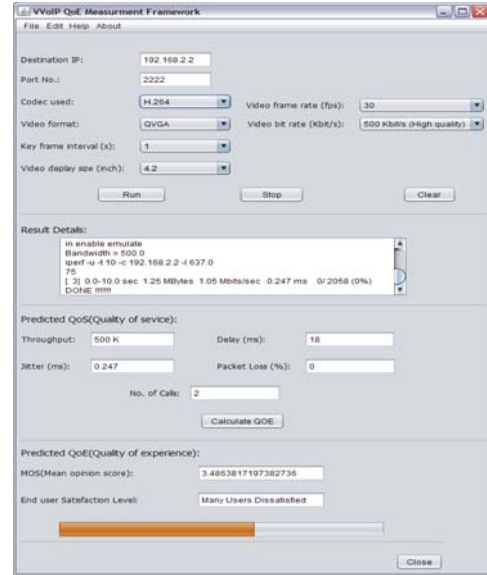


Fig. 6: Screenshot of Video Testing GUI

$$MOS = \begin{cases} 5 & \text{if } PSNR > 37 \\ 0.15 \times PSNR - 0.65 & \text{if } 31 \leq PSNR \leq 37 \\ 0.153 \times PSNR - 0.813 & \text{if } 25 \leq PSNR \leq 31 \\ 0.184 \times PSNR - 1.673 & \text{if } 20 \leq PSNR \leq 25 \\ 1 & \text{if } PSNR < 20 \end{cases} \quad (16)$$

We show sample of our results on two resolutions, QQVGA (160x120) and QVGA (320x240), with frame rates of 15 fps and 25 fps respectively.

PSNR dB	MOS
>37	>5 (Excellent)
31 - 37	4 (Good)
25 - 31	3 (Fair)
20 - 25	2 (Poor)
<20	<1 (Bad)

The comparison is presented in the Figs 7 and 8. The x-axis represents the packet loss rate ranges from 0-6% and the y-axis represent the MOS score of the framework and equivalent PSNR values.

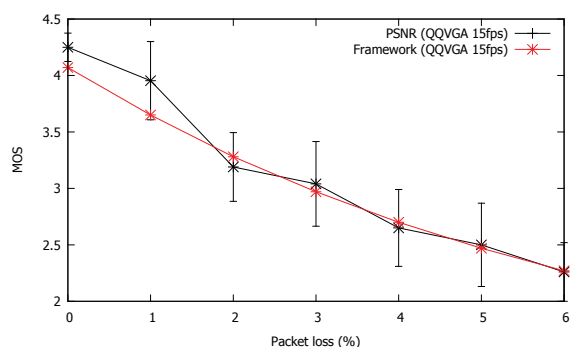


Fig. 7: QQVGA at 15 fps and bitrate of 300 Kbit/s.

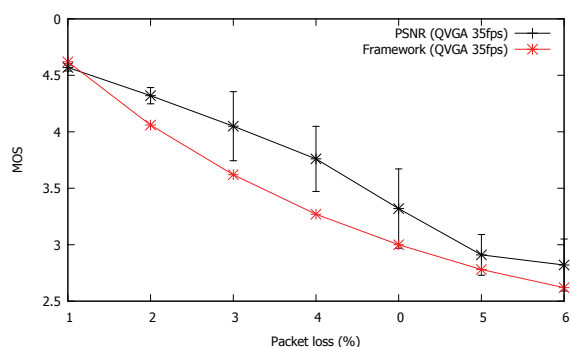


Fig. 8: QVGA at 25 fps and bitrate of 500 Kbit/s.

VI CONCLUSION AND FUTURE WORK

Since processing audio/video sequences is time consuming and computationally intensive, existing objective QoE estimation techniques are not suited for online use. Furthermore, because audio/video codes each have different characteristics it is very difficult to use these techniques to assess in advance which is the codec most appropriate for use giving the prevailing network conditions. To address these limitations we have developed a QoE estimation framework for audio/video that does not require transfer of audio/video sequences or end user involvement. Our experiments show that our framework can achieve acceptable results in comparison to those achieved using the most commonly used industry techniques for audio and video quality testing; PESQ and PSNR respectively. For future work, we are intending to support wideband audio codecs and more video codecs. Moreover, our framework may be improved by online graphical representation for the metrics measured.

ACKNOWLEDGMENT

The authors were supported by Science Foundation Ireland (SFI) grants 07/SK/I1216a and 08/SRC/I1403.

REFERENCES

- [1] ITU-T. P.800: Methods for subjective determination of transmission quality. *ITU-T Recommendation*, 1996.
- [2] ITU-T. Bt.500-10: Methodology for the subjective assessment of quality for television pictures. *ITU-T Recommendation*, 2002.
- [3] ITU-T. P.862: Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs. *ITU-T Recommendation*, 2001.
- [4] ITU-T. P.862.2: Wideband extension to recommendation p. 862 for the assessment of wideband telephone networks and speech codecs. *ITU-T Recommendation*, 2005.
- [5] ITU-T recommendation G.1070. Opinion model for video-telephony applications, April 2007.
- [6] ITU-T. G.107: The E-model, a computational model for use in transmission planning. *ITU-T Recommendation*, 2003.
- [7] C. Jiang and P. Huang. Research of monitoring VoIP voice QoS. In *Proc. 2011 International Conference on Internet Computing & Information Services (ICICIS 2011)*, pages 499–502. IEEE, 2011.
- [8] H.J. Kim and S.G. Choi. Traffic quality monitoring system between different network providers. In *Proc. 12th International Conference on Advanced Communication Technology (ICACT 2010)*, volume 2, pages 1153–1158. IEEE, 2010.
- [9] JM da Silva and RD Lins. Analyzing the QoS of VoIP on SIP in java. In *Proc. 2006 International Telecommunications Symposium*, pages 576–581. IEEE, 2006.
- [10] L. Carvalho, E. Mota, R. Aguiar, A.F. Lima, and J.N. de Souza. An E-model implementation for speech quality evaluation in VoIP systems. In *Proc. 10th IEEE Symposium on Computers and Communications (ISCC 2005)*, pages 933–938. IEEE, 2005.
- [11] Y. Gong, F. Yang, L. Huang, and S. Su. Model-based approach to measuring quality of experience. In *1st International Conference*

- on *Emerging Network Intelligence*, pages 29–32. IEEE, 2009.
- [12] P.C. Hershey, J.M. Pitts, and R. Ogilvie. Monitoring real-time applications events in net-centric enterprise systems to ensure high quality of experience. In *Proc. 2009 IEEE Military Communications Conference (MILCOM 2009)*, pages 1–7. IEEE, 2009.
- [13] P. Calyam, E. Ekici, C. Lee, M. Haffner, and N. Howes. A “GAP-model” based framework for online VVoIP QoE measurement. *Journal of Communications and Networks*, 9(4):446, 2007.
- [14] A.D. Clark, P.D.F. Iee, and et al. Modeling the effects of burst packet loss and recency on subjective voice quality. 2001.
- [15] R.G. Cole and J.H. Rosenbluth. Voice over IP performance monitoring. *ACM SIGCOMM Computer Communication Review*, 31(2):9–24, 2001.
- [16] LCG Lustosa, LSG Carvalho, PHA Rodrigues, and ES Mota. E-model utilization for speech quality evaluation over VoIP-based communication systems. *Proc. 22nd SBRC*, 2004.
- [17] ITU-T. Transmission Impairments due to Speech Processing. Technical Report Recommendation G.113, 2001.
- [18] Haytham Assem, Mohamed Adel, Brendan Jennings, David Malone, Jonathan Dunne, and Pat O’Sullivan. A generic algorithm for mid-call audio codec switching. In *QCMAN 2013 ()*, Ghent, Belgium, May 2013.
- [19] Iperf. <http://iperf.sourceforge.net/>, 2012.
- [20] M. Carbone and L. Rizzo. Dummynet revisited. *ACM SIGCOMM Computer Communication Review*, 40(2):12–20, 2010.
- [21] IBM SUT, 2012.
- [22] J.R. Ohm. *Multimedia communication technology: Representation, transmission and identification of multimedia signals*. Springer Verlag, 2004.