

VoicePerf: A Quality Estimation Approach for No-reference IP Voice Traffic

Péter Orosz and Tamás Tóthfalusi

Abstract— The growing scale of Voice over LTE deployments as well as the wide spread of Internet voice services increase the demand for their user centric service quality monitoring. The primary question in this domain is how a provider can monitor the quality of user experience without executing subjective assessments or applying the common full reference objective quality models. While the former is time and resource consuming and typically performed ad-hoc, the latter requires a reference source, and by processing the voice data it may offend user privacy.

In this paper, we introduce a novel packet-level measurement method (including a novel metric set) to measure network and service quality online. It is done without inspecting the voice payload and needing for the reference voice source. The performance of the proposed model is validated against a full reference voice quality estimation model called AQuA, using real VoIP traffic (generated from various voice samples) in controlled network scenarios.

Index Terms—voice service, performance metrics, VoLTE, VoIP, QoE, voice call quality, PESQ, POLQA

I. INTRODUCTION

PACKET-BASED voice services, due to the improved service quality and the higher performance personal devices, became the dominators of the global voice communication in the last couple of years. The voice-based mobile communication evolved radically, i.e., Voice over LTE (VoLTE) and Voice over IP (VoIP) opened the way towards high definition real-time voice services. This technological evolution raised users' expectation against interactive voice services. Transmitting live voice data over IP networks, which became a popular service of the near past (VoLTE mobile calls and Viber, Whatsapp, Skype, etc. VoIP calls), defines strict real-time criteria, including the control of end-to-end delay, delay variation and packet reordering. From the perspective of the service and network operator, a key management task is to obtain a direct feedback about the quality of the provided voice service. At this point, the most important question is how the provider can monitor the service whether it complies with the raised expectation against user experience.

The operator can monitor call control or voice traffic and analyze the packet-level QoS parameters (e.g., delay, delay variation, packet loss and reordering). While this method reveals basic transmission impairments, does not give a result

that is based on the viewpoint of the voice application. Packet-level analysis of QoS parameters omits the presence and the effect of application level media buffering (i.e., jitter buffer) that directly influences the perceptible call quality.

Alternatively, a provider can analyze the voice waveform using a reference-based objective quality model (e.g., PESQ [1], POLQA [2]). The main drawbacks of the latter algorithms are *a)* offline operation (i.e., post-processing), *b)* requirement for a reference source, and as a fundamental privacy issue, *c)* processing user data, i.e., decoding the voice conversation for quality analysis.

While media contents of music streaming and Internet radio services are typically transmitted over the Transmission Control Protocol (TCP), the interactive real-time services, e.g., VoIP and VoLTE operate over the connectionless User Datagram Protocol (UDP). The key differentiator is the real-time criterion, which TCP, due to its self-clocking and retransmission mechanisms, cannot meet. The higher layer transport protocol commonly used in the mentioned systems is the Real-Time Transport Protocol (RTP) [3], which is an UDP-based protocol carrying the media flow end-to-end. RTP uses sequence numbering and timestamping to relay the time-sensitive media frames over packet-switched networks. The main feature of the protocol is the ability to detect packet losses and to recover orderliness and timing at the receiver side. The latter feature is based on a special timestamp format derived from the voice codec's internal sampling frequency.

Typically, packet-level QoS parameters are measured on the basis of the corresponding IETF RFC proposals [4]–[10]. A common property of the RFC-based methods is that they separate the reorder and loss concepts. In the case of real-time media transmission, we think that it may have several benefits to handle these parameters side by side.

This paper aims to present VoicePerf (Voice Service Performance Measurement Metrics), a new evaluation method for RTP-based real-time flows, to measure the transmission properties from the application's perspective. It operates without inspecting the voice payload (i.e., the private media data) and needing for the corresponding reference voice material.

The algorithm is based on passive measurement and uses a theoretical (reference) timeline combined with a windowing technique. Our measurement system involves five traffic classes and a 3-element metric set, accordingly: *early-arrived loss*, *late-arrived loss* and *not-arrived loss*. Based on the new metric-set, we propose a derivative Mean Opinion Score (MOS)

index that estimates the perceptive quality of the voice session.

II. RELATED WORK

A widely used standardized opinion model for subjective voice quality estimation is the E-model [11], which is an ITU-T standard registered as Recommendation G.107. The standard uses 21 input parameters (e.g., delay, talker echo, loudness, speech distortion) and computes an R-value as output that has a correlation with MOS [12]. Several papers examined the effectiveness of the E-model for voice quality estimation, and found, that it is not accurate enough for estimating subjective quality [13]. It underestimates the voice quality in the case of a longer transmission delay [14] and does not take the burst packet loss into account [15].

A. Takahashi *et al.* [13] proposed a modified E-model for quality estimation. The authors found relationship between network delay and speech distortion. The paper also investigates the problem of terminals' loss rate. Since the jitter buffer implementation can be different in each terminal type, the packet loss rate can be also different and thus a dedicated calibration file (describing the characteristics of the jitter buffer) should be used to assist the model [16].

A. Raake *et al.* [14] presented an improved E-model and extended the original parameters with two new inputs: random loss and burst ratio of the loss. Authors also introduced two packet loss behavioral models: macroscopic loss (speech quality changes over time) and microscopic loss (the effect of packet loss at the decoder side). The prediction model derived from the average packet loss has been adopted in the E-model.

A. Takahashi *et al.* [17] proposed an E-model based method that describes the relationship between delay effects and speech distortion, when both occur at the same time. The authors estimated a subjective MOS from the R-value by transformation. The new MOS estimation model uses the one-way-delay, echo-path loudness rating, noise level, and equipment impairment factor as input parameters. The proposal assumes the speech coding technology and the packet loss rate to be previously known.

Y. Jung and C. Manzano [15] presented a study about the effects of burst packet loss. The authors also found that the E-model does not calculate with the burst packet loss. Accordingly, the proposal defines novel burst-related metrics: burst duration, burst density, gap duration and gap density. Their experimental results showed that the PESQ scores correlate with the above loss parameters. The authors defined multiple loss-range categories for MOS estimation and found that under 7% of packet loss there is no any benefit, in terms of accuracy, to include the burst in the estimation formula.

S. R. Broom [18] found that the system characteristics should be calculated as a degradation value for MOS estimation. The paper concludes that the network loss and jitter values are insufficient for estimating the voice quality. Author introduced a calibration method to measure the characteristics of the VoIP equipment. Using this device specific parameter, a higher correlation rate is achievable.

Y. Ouyang *et al.* [19] proposed an Android App implementation for VoLTE quality estimation. The system

examines the wireless segment of the communication path and uses the POLQA standard for initial calibration. The applied model uses two phases: a training phase and a testing phase. In the training phase, the mobile phone is directly connected to a box to measure MOS values based on reference voice signals. In the testing phase, the client runs in the background and periodically sends network indicators to a database server for further analysis.

D. Luksa *et al.* [20] showed that the choice of the codec (G.711, GSM, Speex, iLBC 20, iLBC 30, and G.729) has no impact in the intelligibility of voice performance and does not influence the objective voice quality.

W. Zou *et al.* [21] proposed a machine learning algorithm for voice quality estimation. The authors used a random forest-based training and assessment method that applies 9 input parameters for the assessment (e.g., UDP length, bitrate, and average packet loss). PESQ scores were used as reference for the analysis of 2,400 degraded voice samples. Test results showed that the correlation between the proposed model and PESQ is higher than between PESQ and the E-model.

Y. Han and G. Muntean [22] present a hybrid call quality assessment model, namely HCQ. The authors collected the benefits of the intrusive (e.g., POLQA, PESQ) and non-intrusive (e.g., E-model) methods and combined them into one solution. The system calculates two MOS values, which are invoked in the final calculation phase. One is the result of the online model and the other one is estimated by an offline reference-based model. The offline MOS calculation is based on small voice segments, where the system records a few seconds and estimates its quality with PESQ. The drawback of this solution is that the recorded voice is sent through the network for analysis, which is not a viable option in many scenarios. Authors recommend a reduced reference method for the offline calculation.

N. Majed *et al.* [23] analyzed delay-based measurement methods in VoLTE environment. The authors found that 3GPP test methods (e.g., IPDV) do not model the behavior of jitter-buffers in real VoLTE systems and they presented a set of improvements to achieve a higher correlation.

M. Abareghi *et al.* [24] proposed an improved ITU-T standard, namely P.563 [25]. This standard incorporates a non-intrusive model for voice quality assessment, but it is not appropriate for VoIP calls. The authors presented a new distortion class for a proper network condition detection.

D. E. Conway [26] proposed a passive measurement method that incorporates the offline quality estimation model called PESQ and a media payload injection technique to replace the original user media data with a test media source. Using payload replacement, author avoided any privacy issues raised from decoding user's media payload. However, the proposed method does not reflect to the properties of the application's playout buffer and therefore the real data loss present at the input of the voice decoder is not incorporated in the result of the quality assessment.

L. Sun and E. C. Ifeachor [27] presented a new passive (non-intrusive) methodology for developing accurate models to predict voice quality for IP-based voice services. Based on the

methodology, authors introduced regression models for predicting conversational voice quality for four common voice codecs: G.729, G.723.1, AMR, and iLBC. By using real VoIP traces authors showed that the prediction accuracy of the generated models is close to the one of the combined ITU PESQ/E-model method.

J. C. w. Lin *et al.* [28] introduced a new parametrical neural network-based model to estimate the voice quality in Voice over IP systems. The work enhances the model presented in [27]. The main contribution of their work is the statistics-based packet loss evaluation. The proposed method is more efficient computationally since it does not require Markov model mapping.

Let us note that both [27] and [28] lack considering the effect of packet reordering and packet losses derived from the overrun of the playout buffer (i.e., early-arrived loss due to a traffic burst).

In contrast to previously published works, our proposed method distinguishes the packet loss events and defines three loss categories and applies a time-windowing technique to model the endpoint's playout buffer behavior. Based on the new metric-set, we propose an estimation model to get a derivative MOS index. Additionally, it is enhanced with a new method to detect the pace of the speech (dynamic or slow) when voice activity detection (VAD) is present between the endpoints. This identification supports the introduced quality model to get an estimation with higher correlation to the perceptive quality.

III. THE PROPOSED METRIC CALCULATION MODEL

The proposed VoicePerf method is based on passive network measurement, in which the original packets are not altered through extra header information, and the voice payload is not decoded at all. Accordingly, it does not induce neither additional network overhead, nor privacy issues.

Our method invokes only two protocol metadata, i.e., the sequence number to measure packet orderliness and the packet timestamp to analyze timeliness.

A. Introducing the packet-level VoicePerf metrics

First of all, to measure the characteristics of real-time flows from the application's viewpoint, we have to examine the operation of the RTP protocol. RTP assists media delivery by recovering order and time information at the receiver side. These protocol features enable a scheduled processing (decoding) of media frames delivered through the network. Accordingly, each RTP header contains a sequence number and a timestamp value. While the sequence number is incremented by one for each packet and used to detect loss or reordering, the timestamp carries codec related temporal information, and is used to schedule the arrived packets in time. For voice services, RTP timestamp is incremented in every packet by the number of audio samples it carries. For example, when a packet contains 20 ms of audio sampled at 8 kHz, the timestamp is incremented by 160 in each RTP packet (see Fig. 1). Let us note that the initial value of the sequence number is random within the 16-bit range.

Let S_i be the sequence number of the i^{th} arrived packet, and

let P_i be the packetizing period of the currently used codec.

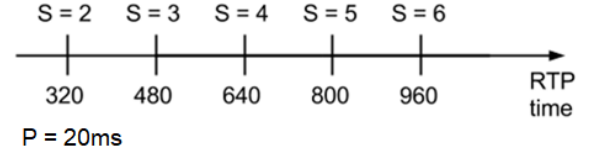


Fig. 1. Timestamp increment in RTP packets

Based on this scheduling scheme, VoicePerf applies a reference timeline, on which current time is incremented by the delta arrival time between two consecutive packets. A localtime-based timestamping mechanism is used for the delta time calculation. The reference timeline will be used to classify an incoming packet based on its time domain properties.

Let t_i be the arrival time and let d_i be the delta time of the i^{th} packet from the previously arrived packet. Let Ct be the current time that represents a time relative to the beginning of the measurement. When a packet arrives, the delta time is calculated and added to current time Ct (Fig. 2).

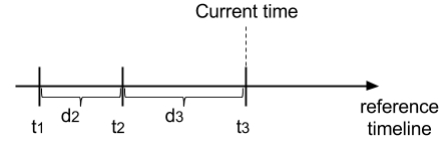


Fig. 2. Reference time incremented by delta arrival times

Besides the calculation of the reference time, VoicePerf applies windowing to perform classification and to calculate the metrics. Our proposed model handles five time-windows that are defined by six reference points in time. The reference points and thereby the time-windows are re-calculated for each arrived packet in real-time and fitted on the reference timeline to determine the performance metrics (see Fig. 3 and 4). The size of a time window accommodates to the size of a voice frame (20 ms, typically).

Let Rp_{ij} be the j^{th} reference point of the i^{th} arrived packet, where $1 \leq j \leq 6$. The method assumes P_i to be known in each calculation cycle. When the RTP stream involves multiple payload types (i.e., multiple codec and sampling period combinations), P_i should be properly synchronized to the arrival events. The rate could be preset, if it is previously known from the codec properties, or could be indirectly measured in the initial phase of the algorithm, if it is unknown. (The initial phase will be discussed later in this section.) Let us note that the resolution of P_i should be equal of the resolution of Ct .

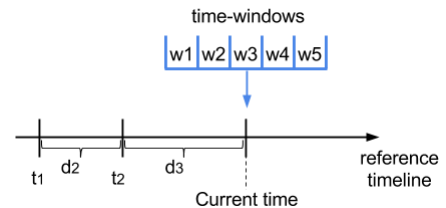


Fig. 3. Time-window fitting on the reference timeline

Let T be a threshold value that defines a global time interval for the packet classification. Its value is determined by the playout buffering scheme of the voice application. T is practically an integer value, and $T \geq 2$. The T threshold makes the method scalable for any playout buffer size.

Using the previously defined T parameter, Rp_{ij} reference points are calculated by formulas (1-6), respectively:

$$Rp_{i1} = (S_i \times P_i) - ((T + 0.5) \times P_i) \quad (1)$$

$$Rp_{i2} = (S_i \times P_i) - (1.5 \times P_i) \quad (2)$$

$$Rp_{i3} = (S_i \times P_i) - (0.5 \times P_i) \quad (3)$$

$$Rp_{i4} = (S_i \times P_i) + (0.5 \times P_i) \quad (4)$$

$$Rp_{i5} = (S_i \times P_i) + (1.5 \times P_i) \quad (5)$$

$$Rp_{i6} = (S_i \times P_i) + ((T + 0.5) \times P_i) \quad (6)$$

The reference points define five time-windows (w_k , where $1 \leq k \leq 5$) based on the following intervals:

$$Rp_{ik} \leq w_k < Rp_{ik+1}$$

Each of the three middle time windows (w_2 , w_3 and w_4) defines a fixed interval, while the two outer (w_1 , w_5) windows represent an application specific interval. Windows w_2 , w_3 and w_4 are determined by the increment value (resolution) of the RTP timestamp. Both w_1 and w_5 have an application-specific size, which could be increased by the T threshold value.

Rp_{ij} reference points are updated for each arrived packet and time-windows are slid accordingly. Then they are fitted on the reference timeline to perform classification and to calculate the performance metrics. Figure 4 presents an example calculation: $T = 2$, $S = 5$, $P = 20$ and $Ct = 104$.

VoicePerf takes the arrived sequence number S_i and the packetizing period P_i to determine the relevant time-windows w_1 - w_5 and the expected arrival interval for the current voice frame (see Formulas 1-6). The classification categories are defined via the corresponding time windows. Classifying a packet to a pre-defined category is a 1:1 association, where categories correspond to performance metrics.

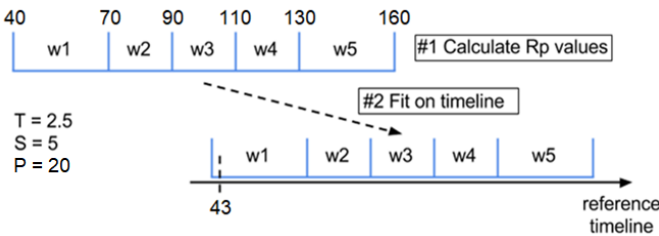


Fig. 4. Time windows in the case of $S=5$, $P=20$, $Ct=43$, $T=2.5$

Our method has an initialization phase followed by an assessment phase. During the initial phase, the reference timeline and the packetizing period should be properly adjusted. A reference timer is typically a common local clock source (e.g., TSC, HPET), which is used to timestamp the arrived packets. However, the reference timeline is initialized by the $S_0 \times P_0$ product. If reordering or loss occurs in the init phase, the reference timeline must be re-initialized by the next in-order

packet. The method maintains the initialization phase until the packetizing period is fixed and no out-of-order packet arrives for at least a $(4 + 2T)P_i$ period. This criterion grants that the values of the Rp_{ik} reference points will be higher than the initial value of the timeline. As soon as the above conditions are met, the method steps into the assessment phase. The assessment phase relies on the previously introduced time-windows, applying the categories and the corresponding performance metrics.

B. Categorization of the arrived packets - Loss Category Types

VoicePerf defines loss categories for the arrived packets (Early arrived loss – EAL, Late arrived loss - LAL), which are differentiated by the time-windows.

Since voice codecs and thus RTP operate with time frames, the decoding algorithm always has to process a sequence of voice frames. When a packet comes earlier than it is expected, the media application stores its payload in the playout buffer. When a packet with an early arrival faces a filled buffer, the application may discard its payload (i.e., the voice frame) despite the packet arrived to the endpoint. This is because its playout time is far from the current time window. VoicePerf defines the early-arrived loss metric for the representation of media frames arrived too early. Using the windowing technique, the packet is counted as early arrived loss when $Ct < Rp_{i1}$.

Similarly, we can differentiate the late arrived loss event. When a packet arrives too late from its expected interval, the application may handle the missing time information with different techniques, because the received audio information is irrelevant in time. VoicePerf counts the i^{th} packet as late arrived loss when $Ct > Rp_{i6}$.

When Ct of the i^{th} arrived packet is greater than or equal to Rp_{i1} or less than or equal to Rp_{i6} , the packet is categorized as time-window offset. This is because the application may use the carried voice frame and places it in the playout buffer. VoicePerf also counts each packet within a measurement period with the appropriate time-window counter. The variance of arrivals within the time-windows can be an early indication of a transmission timing problem.

C. Representation of the categories and time windows

Based on the time-windows, Ct exactly defines the relationship between the packet and the playout buffer usage. Generating a histogram from the metric counters represents a graphical feedback about the network conditions besides buffer usage from the application's perspective. Using the sliding window technique, the feedback histogram represents a live chart. Figure 5 show a synthesized example.

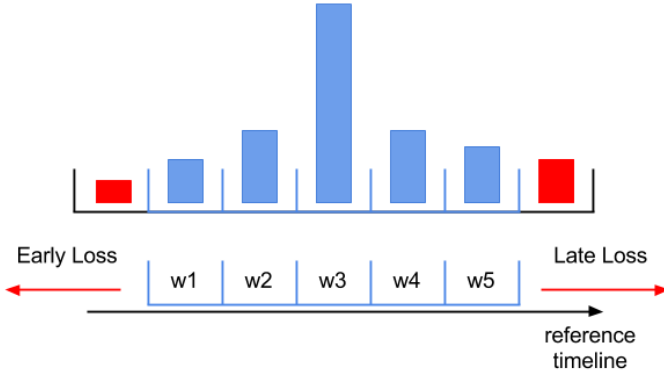


Fig. 5. Live chart about the packet classification

IV. ESTIMATING PERCEPTUAL QUALITY OF THE VOICE CALL USING THE PROPOSED METRICS

By introducing a category quality indicator (CQI) for each loss-category as a VoicePerf specific weight, we can calculate a new QoE estimation index on the common P.800 MOS-scale. This quality index represents the overall performance of the call session. It is important to note that by calculating QoE the primary aim is to provide an indication how a network path complies with the requirement of delivering voice traffic end-to-end. Typically, a low-level degradation of the transmission, in terms of timing, can be eliminated by the receiver side jitter buffer, and therefore the perceived service quality may be unaffected. Nevertheless, such low-level impairments can be forerunners for a negative trend within the network path (i.e., evolving congestion) that will degrade user experience in the very near future.

A. Slow and Dynamic speech types

During the correlation procedure we differentiated two speech types: slow and dynamic.

To differentiate the slow and the dynamic speech types, we have to briefly introduce the voice activity detection (VAD) method commonly used in IP-based voice services. VAD is an important technique to reduce the bit rate of speech in VoLTE and VoIP applications. The method detects the silence periods of the speech and indicates the start and the end of the silence in the RTP flow. During a silence period, sender side does not transmit voice packets, meanwhile the decoder at receiver side generates comfort noise for the listener. This is a frequent scenario, when the caller or the callee listens and does not say anything. These silence periods should be recognized for a QoE estimation and the slow or dynamic CQI values should be differentiated based on the VAD properties.

The primary question about differentiating slow and dynamics speeches is how the speech type can be identified without inspecting the voice payload. Accordingly, we proposed a new real-time method that relies on the threefold relationship between speech type, non-VAD/VAD ratio and alteration of voice packet rate. The packet rate is constant (50 pps, typically) as long as the voice coder is not in VAD state, we can detect the ratio of non-VAD and VAD states as per time unit. Since in VAD mode the packet rate is significantly

reduced by the codec, we propose a method to identify the speech type based on the effective packet rate (corrected by the measured loss) versus the maximum packet rate (i.e., the constant 50 pps packet rate of the non-VAD mode). Real life speech examples are shown in Fig. 6 and 7.

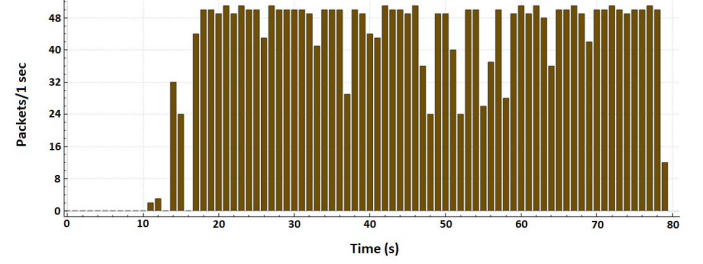


Fig. 6. Packet rate pattern for dynamic speech

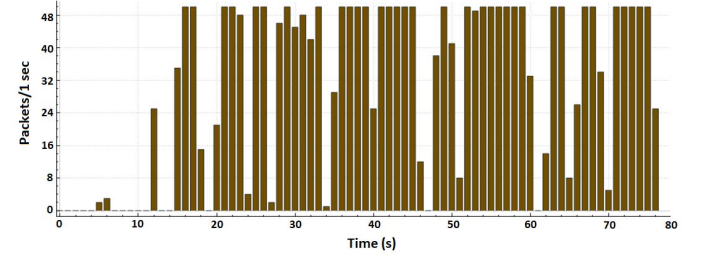


Fig. 7. Packet rate pattern for slow speech

TABLE I
Speech type versus effective packet rate (pps)

Type of speech	#1	#2	#3	Average
Male - Dynamic	0.877	0.872	0.904	0.88433
Male - Slow	0.726	0.767	0.653	0.71533
Female - Dynamic	0.883	0.869	0.893	0.88167
Female - Slow	0.722	0.619	0.732	0.691

According to the analysis of our 12 voice samples (Table I), the average of the effective packet rate is 0.883×50 pps for the dynamics speeches and 0.703 for the slow ones. Accordingly, we set up a 0.8 threshold to automatically separate the two speech types.

B. The measurement environment

To determine the CQI weights, we generated several audio files with various impairment parameters. Our measurement setup contained two laptops (laptop-A as caller and laptop-B as callee, see Fig. 8) and an OpenWrt [33] router to emulate network behavior. We applied the NetEm [34] (Network Emulator) Linux kernel module on the router to drop, delay and reorder the transmitted RTP packets. We used Ekiga [35] as a VoIP client on both laptops to perform IP-based voice calls. While we captured the RTP packets losslessly, we also recorded the speech waveform with Audacity [36]. As voice references, we chose four independent samples from audiobooks: a slow (calm) male speech, a slow female speech, a dynamic male speech, and a dynamic female speech. We replayed the audio files and streamed into the VoIP call. In the Ekiga client, we applied AMR-WB as the default codec type and set up a jitter buffer of 100 ms.

Since the input jitter buffer eliminates the effect of a reordered packet with a relatively low sequence offset and grant an ordered playout, a reordered packet with a lower delay variation than the buffer capacity does not result voice quality degradation. However, a larger sequence offset drives to receiver side packet elimination, i.e., data loss. We checked this statement with replayed audio files.

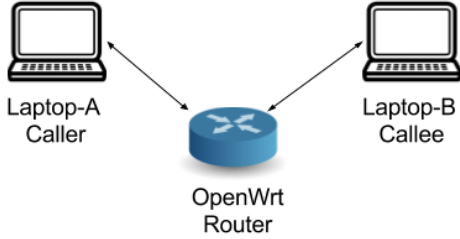


Fig. 8. Laboratory setup for audio test file generation

In contrast to measure reordering and packet loss independently, we analysed both side by side to provide a higher accuracy estimation of the voice quality degradation. We generated test audio files using the emulator system configured with 2, 5, 10, 20, 30 and 40 percent of packet loss, respectively. We applied the related network QoS scenarios on each of the four reference speeches. We calculated the MOS value for each output voice file using the AQuA [29] analyzer tool, and also measured its packet-level loss metrics according to the proposed VoicePerf method.

In our study, we profiled the early loss, late loss and not-arrived loss categories. We observed that a late-arrived voice packet results in a same data loss scenario than a not-arrived loss packet. Both cases imply the same type of data loss at the input of the voice decoder. Accordingly, we investigate the late-arrived and not-arrived loss categories as a combined not-arrived loss category in the rest of the paper.

Based on the measurement results, we found that the characteristics of the early-loss differs from that of the not-arrived loss (see Fig. 9 and 10). Accordingly, the MOS index should be determined by unique functions.

C. Weight calculation for network and speech quality KPIs

Using the calculated CQI indicators, we found correlation between VoicePerf metrics and service QoE. To define an appropriate function for calculating the CQIs, we profiled the loss categories by applying the Audio Quality Analyzer software (AQuA) [29] from Sevana.

There are several test methodologies to measure the perceptual quality degradation in a voice material. One of the well-known algorithms is the Perceptual Evaluation of Speech Quality (PESQ), which is standardized as ITU-T P.862 recommendation [1]. PESQ applies a full reference (FR) model to estimate the quality degradation and calculate a Mean Opinion Score (MOS) [30] value. FR means that the algorithm is based on a reference speech signal, which is compared to the degraded waveform.

A newer ITU-T standard is the Perceptual Objective

Listening Quality Analysis (POLQA) model [2], which applies new methods for wideband and super-wideband voice signals.

AQuA is an easy-to-use voice quality tester. It also implements an FR model, which provides quality analysis between audio files. The software calculates an estimated P.800 MOS value, as a classic indication of perception, and a quality percentage value, both derived from the waveform. This latter, i.e., the quality percentage calculation procedure, is based on the analysis of spectrum vibration, energy distortion and other waveform specific parameters. AQuA uses a different perceptual model than PESQ to reveal more information about the loss of voice quality. This is because there are cases of degradation that PESQ fails to detect [31]. For giving a comparative summary, Sevana collected the features of the well-known speech quality calculation methods, i.e., PESQ, POLQA, and AQuA [32].

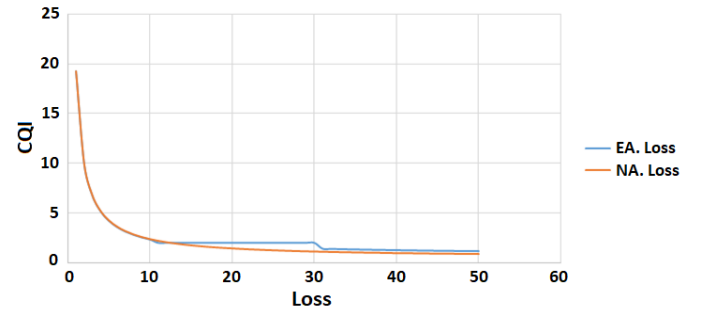


Fig. 9. CQI weight functions of dynamic speech

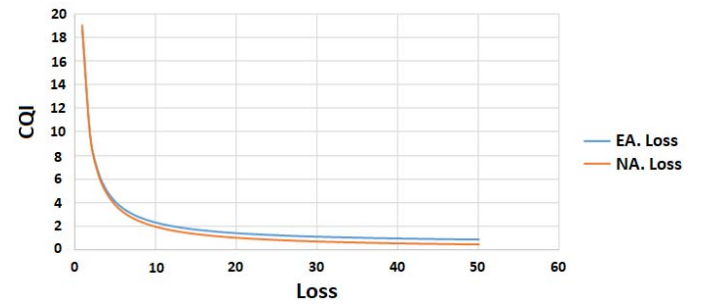


Fig. 10. CQI weight functions for slow speech

To get general weight functions for the QoE estimation of call quality, we applied regression analysis using curve estimation in IBM SPSS [37]. We used the inverse model for both slow and dynamic speeches. The resulted linear regression function on the not-arrived loss weights show a correlation of 99.8% in the case of dynamic speech, and 100% in the case of slow speech. Considering the early-arrived loss, the regression analysis reveals a correlation of 98.2% for the dynamic speech and 99.6% for the slow one.

Using the measured loss value as explanatory variable, the weights for different speech types can be determined by unique predictor functions. Based on the outcomes of the inverse model-based regression analysis, we defined two weight (predictor) functions (7, 8) for the slow speech segments ($h_{NAL}(p)$ for not-arrived loss and $h_{EAL}(p)$ for early loss), and two others (9, 10) for the dynamic speech segments ($g_{NAL}(p)$ for not-arrived loss and $g_{EAL}(p)$ for early loss). The p explanatory variable is the packet loss rate for the given time period.

$$h_{NAL}(p) = 0.093 + \frac{0.18923}{p} \quad (7)$$

$$h_{EAL}(p) = 0.48 + \frac{0.182}{p} \quad (8)$$

$$g_{NAL}(p) = 0.453 + \frac{0.18796}{p} \quad (9)$$

$$g_{EAL}(p) = \begin{cases} 0.453 + \frac{0.18796}{p}, & p \leq 0.1 \\ 2, & 0.1 \leq p \leq 0.3 \\ 0.747 + \frac{0.206}{p}, & p > 0.3 \end{cases} \quad (10)$$

In the case of late-arrived loss, we use the predictor functions (7) and (9) for determining the appropriate CQI weights.

This difference suggests that the punishment of the early- and the late- or not-arrived loss should not be equal.

Let us note that each of the weight functions (7)-(10) assumes a non-zero p value. When the p loss value is zero, CQI calculation is omitted.

D. Burst-loss measurement

Based on the experience of related researches [15], the burst packet loss is also a MOS-reducing factor. We found in our measurement samples that 8-10 consecutive packet loss events, which spans about a 160-200 ms time interval, result in a perceptible crackling sound. The number of crackling events is an annoying factor, which reduces the quality of the speech. It should be handled for a proper assessment algorithm.

Let LDC be the loss discontinuity counter and LDT be a loss discontinuity threshold. LDC is incremented when a loss event occurs (early-arrived, late- or not-arrived loss), and is reset to zero when it equals to LDT or when the packet arrived in one of the w_1 - w_5 time windows. Let LDE be the loss discontinuity event, which is incremented by any LDC reset event.

TABLE II
SUMMARY TABLE FOR LDE WEIGHTS

Type of speech	LDE < 10	10 < LDE < 15	15 < LDE
Dynamic	0.012	0.006	0.002
Slow	0.002	0.004	0.008

To achieve a better correlation with MOS, we applied the calculated LDE counter (the number of crackling sound events) as a degradation parameter in our QoE estimation formula. We defined three LDE-weight categories based on the measurement samples (see Table II) to specify a weighted LDE value for (12).

E. MOS Estimation

To estimate the MOS value from the VoicePerf loss values, the three loss categories (i.e., early loss, late loss, not-arrived

loss) have to be weighted by the previously defined CQI predictor functions. For a higher correlation, the LDE value should be also weighted. Let $wLDE$ be the proper weight for the LDE counter based on the results in Table II.

Let P_c be the rate of the given category counter versus the total RTP packets sent, where c means the category (NAL, EAL, and LAL) and $0 \leq P_c \leq 1$.

The estimated QoE value is calculated by (11)-(14). Let P_{SL} be the sum of P_{NAL} and P_{LAL} . Let $f(p, st)$ be the function, used in (11), where p is the packet loss and st is the speech type, that returns the appropriate CQI function ($h_{NAL}(p)$, $h_{EAL}(p)$, $g_{NAL}(p)$ or $g_{EAL}(p)$, respectively) based on the speech type and the corresponding loss categories.

$$M1 = \frac{(1 - P_{SL} \times f(P_{SL}, st)) + (1 - P_{EAL} \times f(P_{EAL}, st))}{2} \quad (11)$$

$$M2 = wLDE \times LDE \quad (12)$$

$$M3 = M1 - M2 \quad (13)$$

$$VoicePerfMOS = 5 \times M3 \quad (14)$$

V. VALIDATION

To validate our voice quality assessment model (14) in laboratory environment, we implemented the proposed algorithm in C++, and applied it on sample capture files.

The measurement setup for the validated sample generation was similar to Section III.C.1. We used two laptops as caller and callee (see Fig. 8), and an OpenWrt router between them to emulate network impairments. We also used Ekiga [35] for the VoIP call, and Audacity [36] to redirect and record the audio files. We generated thirty-four audio files as test cases derived from male and female reference speeches (slow and dynamic types). We selected eight different new voice samples (not overlapping with ones used for the regression analysis) as reference from audiobooks. For each test case generation, we chose one of the eight reference speech and messed up the network properties. To induce early loss, not arrived loss, late loss and burst loss events, we activated and deactivated alternately one of the following NetEm [34] kernel module parameters (network delay, reordering and loss) during the call:

- 400 ms delay, 20% reorder with 80% correlation
- 400 ms delay, 20% reorder with 80% correlation, 10% loss
- 600 ms delay, 20% reorder with 80% correlation, 2% loss
- alternating network delay (0 / 600 ms), 50% reorder with 80% correlation
- 200 ms delay, 50% reorder with 80% correlation
- alternating reorder percentage (0% / 50%)
- alternating reorder percentage (0% / 50%) and alternating network delay (0 / 200 ms)
- alternating reorder percentage (0% / 90%) with 50% correlation

- 5% continuous loss

We applied the AQuA [29] analyzer tool to calculate the MOS values for each speech sample. To determine the accuracy of our VoicePerf quality assessment model, we calculated the window-specific counters and the different loss category counters per speech sample and applied the VoicePerf MOS formula (14).

Fig. 11 represents the correlation between AQuA MOS and VoicePerf MOS values, and also shows the absolute errors. The validation results reveal that the absolute error range is between 0.005 and 0.5 MOS, and the average delta MOS is 0.144. To conclude the observations, the difference between AQuA and VoicePerf MOS in 80% of test cases is less or equal to 3%, in 85% of the cases is less or equal to 5%, and in 100% of the cases is less or equal to 10%.

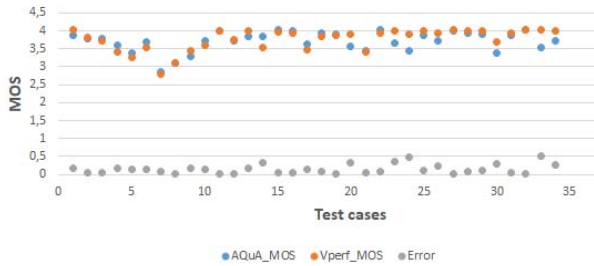


Fig. 11. Validation results: correlation between AQuA and VoicePerf MOS values

The results show that the average delta percent (Table III), which comes from the difference between AQuA [29] MOS and VoicePerf model based MOS values, is less than 2.6% in the case of dynamic speech, and less than 3.6% in the case of slow speech. The slow speech test cases frequently contain 1-3 seconds long silence blocks. Table III shows the summary of the validation results.

TABLE III
SUMMARY OF MOS ASSESSMENT VALIDATION

Type of speech	Maximum delta MOS	Average delta MOS
male dynamic	0.32 (6.4%)	0.13 (2.6%)
male slow	0.34 (6.8%)	0.18 (3.6%)
female dynamic	0.47 (9.4%)	0.12 (2.4%)
female slow	0.5 (10%)	0.15 (3%)

VI. CONCLUSION

The evolution of HD voice-based mobile communication expects several transmission parameters to be kept under tight control to grant reliable and high-quality voice services. It necessitates a user centric service quality monitoring of VoLTE and VoIP applications in real-time.

In this paper, we proposed a measurement method called VoicePerf that contributes three novelties in monitoring the quality of a real-time voice call. i) a new classification-based

metric system for RTP-based voice traffic (early-arrived loss, late-arrived loss and not-arrived loss). ii) We introduced a derivative MOS index that can be calculated for a call session from the measurement results. iii) Additionally, we proposed a new method to identify the pace of the speech (dynamic or slow) as long as voice activity detection (VAD) is present between the endpoints. This identification supports the introduced quality model to estimate the perceptive quality with higher accuracy.

In contrast to previously published works, our proposed method distinguishes the packet loss events and defines three loss categories and applies a time-windowing technique to model the endpoint's playout buffer. Our approach to determine service quality is based on the analysis of media data availability inside the endpoint's playout buffer that is a key element of the media delivery chain. However, instead of analyzing media data directly at the endpoint's playout buffer, we modelled its operation in our independent measurement system using a time windowing method. The main advantage of our proposal (i.e., an integrated reorder-loss calculation) is the ability to determine the status of the arrived IP/RTP packet from the application's viewpoint. Based on these statistics, we get a cross-layer overview that can be applied for estimating the quality of user experience.

We validated our MOS estimation method against a full-reference voice quality estimation model called AQuA with 8 different voice materials (4 male and 4 female samples) in many different network scenarios. We found high correlation between VoicePerf MOS and AQuA MOS in all scenarios. The validation results showed that applying our method (i.e., CQI weights, three types of loss metrics, loss burst measurement, and speech type detection), the QoE could be estimated with an average of 2.6% error ratio for dynamic speeches and 3.6% error ratio for slow ones.

VII. ACKNOWLEDGEMENT

We would like to thank Sevana Inc. for the academic license of the AQuA voice quality analyzer software.

REFERENCES

- [1] "Perceptual Evaluation of Speech Quality (PESQ)", Online, available: <http://www.itu.int/rec/T-REC-P.862>, viewed on 10/09/2019
- [2] "Perceptual Objective Listening Quality Assessment", Online, available: <https://www.itu.int/rec/T-REC-P.863>, viewed on 10/09/2019
- [3] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A transport Protocol for Real-Time Applications", IETF RFC 3550, Online, available: <https://tools.ietf.org/html/rfc3550>, viewed on 10/09/2019
- [4] A. Morton, L. Ciavattone, G. Ramachandran, S. Shalunov, J. Perser, "Packet Reordering Metrics", IETF RFC 4737, Online, available: <https://tools.ietf.org/html/rfc4737>, viewed on 10/09/2019
- [5] A. Jayasumana, N. Piratla, T. Banka, A. Bare, R. Whitner, "Improved Packet Reordering Metrics", IETF RFC 5236, Online, available: <https://tools.ietf.org/html/rfc5236>, viewed on 10/09/2019
- [6] A. Morton, B. Claise, "Packet Delay Variation Applicability Statement", IETF RFC 5481, Online, available: <https://tools.ietf.org/html/rfc5481>, viewed on 10/09/2019
- [7] C. Demichelis, P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", IETF RFC 3393, Online, available: <https://tools.ietf.org/html/rfc3393>, viewed on 10/09/2019

- [8] IETF IPPM Reordering draft, Online, available: <https://tools.ietf.org/html/draft-ietf-ippm-reordering-13>, viewed on 10/09/2019
- [9] V. Raisanen, G. Grotefeld, A. Morton, "Network performance measurement with periodic streams", IETF RFC 3432, Online, available: <https://tools.ietf.org/html/rfc3432>, viewed on 10/09/2019
- [10] A. Morton, G. Ramachandran, G. Maguluri, "Reporting IP Network Performance Metrics: Different Points of View", IETF RFC 6703, Online, available: <https://tools.ietf.org/html/rfc6703>, viewed on 10/09/2019
- [11] "The E-Model, a computational model for use in transmission planning", 2005, ITU-T Rec. G.107.
- [12] L. Ding, R.A. Goubran, "Assessment of effects of packet loss on speech quality in VoIP", The 2nd IEEE International Haptic, Audio and Visual Environments and Their Application, 2003
- [13] A. Takahashi, H. Yoshino, N. Kitawaki, "Perceptual QoS Assessment technologies for VoIP", IEEE Communications Magazine, vol. 42, 2004, pp. 28-34.
- [14] A. Raake, "Short- and Long-Term Packet Loss Behavior: Towards Speech Quality Prediction for Arbitrary Loss Distributions", IEEE Transactions on Audio, Speech, and Language Processing vol 14., 2006, pp. 1957-1968.
- [15] Y. Jung, C. Manzano, "Burst packet loss and enhanced packet loss-based quality model for mobile voice-over Internet protocol applications", IET Communications vol. 8., 2014, pp. 41-49.
- [16] S. Broom and M. Hollier, "Speech Quality Measurement Tools for Dynamic Network Management", MESAQIN 2003
- [17] A. Takahashi, A. Kurashima, H. Yoshino, "Objective Assessment Methodology for Estimating Conversational Quality in VoIP", IEEE Transactions on Audio, Speech, and Language Processing vol. 14, 2006, pp. 1984-1993.
- [18] S. R. Broom, "VoIP Quality Assessment: Taking Account of the Edge-Device", IEEE Transactions on Audio, Speech, and Language Processing vol. 14., 2006, pp. 1977-1983.
- [19] Y. Ouyang, T. Yan, G. Wang, "CrowdMi: Scalable and Diagnosable Mobile Voice Quality Assessment Through Wireless Analytics", IEEE Internet of Things Journal vol. 2., 2015, pp. 287-294.
- [20] D. Luksa, S. Faaht, M. Krhen, "Sound quality assessment in VOIP environment", 37th International Convention on Information and Communication Technology, 2014
- [21] W. Zou, F. yang, X. Li, "A Packet-Layer Quality Assessment System for VoIP Using Random Forest", IEEE International Conference on Computer and Information Technology, 2014
- [22] Y. Han, G. Muntean, "Hybrid real-time quality assessment model for voice over IP", IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, 2015
- [23] N. Majed, S. Ragot, X. Lagrange, A. Blanc, "Delay and Quality Metrics in Voice over LTE (VoLTE) Networks: an End-Terminal Perspective", International Conference on Computing, Networking and Communications, 2017
- [24] M. Abareghi, M. M. Homayounpour, M. Dehghan, A. Davoodi, "Improved ITU-P.563 Non-Intrusive Speech Quality Assessment Method for Covering VOIP Conditions", 10th International Conference on Advanced Communication Technology, 2008
- [25] "Single-ended method for objective speech quality assessment in narrow-band telephony applications", ITU-T Rec. P.563, May 2004.
- [26] A. E. Conway, "A passive method for monitoring voice-over-IP call quality with ITU-T objective speech quality measurement methods," 2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No.02CH37333), 2002, pp. 2583-2586 vol.4.
- [27] Lingfen Sun and E. C. Ifeachor, "Voice quality prediction models and their application in VoIP networks," in IEEE Transactions on Multimedia, vol. 8, no. 4, pp. 809-820, Aug. 2006.
- [28] J. C. w. Lin and P. Fournier-Viger, "Employing the neural networks to parametrically assess the quality of a voice call," 2016 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Montreal, QC, 2016, pp. 1-5.
- [29] Sevana, "AQuA – Audio Quality Analyzer", Online, available: <http://sevana.biz/products/aqua/technology/>, viewed on 10/09/2019
- [30] "Mean Opinion Score (MOS) terminology", Online, available: <https://www.itu.int/rec/T-REC-P.800.1-200303-S/en>, viewed on 10/09/2019
- [31] Microtronix SYSTEMS LTD, "Perceptual Evaluation of Speech Quality (PESQ)", Online, available: <http://www.microtronix.ca/pesq.html>, viewed on 10/09/2019
- [32] Sevana, "PESQ, POLQA, AQUA, Feature Comparison Table", Online, available: <http://sevana.biz/pesq-polqa-aqua/>, viewed on 10/09/2019
- [33] OpenWrt firmware, Online, available: <https://openwrt.org/>, viewed on 10/09/2019
- [34] Linux kernel, Online, available: <http://www.kernel.org>, viewed on 10/09/2019
- [35] Ekiga software, Online, available: <http://www.ekiga.org/>, viewed on 10/09/2019
- [36] Audacity software, Online, available: <http://www.audacityteam.org/>, viewed on 10/09/2019
- [37] IBM Analytics, "IBM SPSS Software", Online, available: <https://www.ibm.com/analytics/us/en/technology/spss/>, viewed on 10/09/2019