Speech Quality Assessment Using Computational E-Model and its Implementation

M. Voznak, F. Rezac and J. Rozhon

***Abstract*—**The paper deals with speech quality assessment in a non-intrusive computational E-model and its implementation in the BESIP (Bright Embedded Solution for IP telephony) project. Main goal of the BESIP project is to develop an advanced open-source communication platform for IP telephony encompassing security and monitoring modules, the whole concept is implemented in OpenWRT distribution for embedded devices with minimal HW requirements. The idea of speech quality assessment in E-model is based on computing the estimated quality from knowledge of transmission path parameters and their impact on the original quality degradation. In order to compute an estimated speech quality in monitoring module from available information which we are able to determine in BESIP, we applied several simplification in E-model. Finally, we implemented the proposed approach into Monitoring module of the BESIP and compared the computed MOS with values gained by an objective intrusive PESQ method (Perceptual Evaluation of Speech Quality). The achieved results proved the applicability of our approach. In this paper, we describe in detail the estimated speech quality computation in simplified E-model, its implementation and verification.

***Keywords*—**E-model, PESQ, Speech quality assessment, Monitoring module, MOS.

I. Introduction

THE speech quality assessment becomes crucial issue of telecommunication operators with transition from TDM to IP telephony. According to the method of assessment, speech quality evaluation methodologies can be subdivided into subjective methods and objective methods [1], [2]. To evaluate speech quality, listeners express their opinions on the quality of the speech samples in terms of five categories: excellent, good, fair, poor and bad, with a corresponding integer score: 5, 4, 3, 2, and 1, respectively. The ratings are averaged and the result is usually known as MOS (Mean Opinion Score) [2], [3]. Objective methods have been developed to fill a need to produce a good estimate of subjective MOS in recent years. These methods can be classified into two categories: intrusive or non-intrusive, based on whether a reference speech is needed or not. This article is organized as follows. The intrusive and non-intrusive approach is discussed in the section 2, next section explains a computational E-model. Section 4 describes our simplification of E-model and its practical implementation. In the section 5, we describe the used testbed for verification and we compare the computed values from our monitoring system with results gained by an objective PESQ method.

Manuscript received February 10, 2013. The research leading to these results has received funding from the Ministry of Education of the Czech Republic within the project LM2010005.

M. Voznak is an Associate Professor with Dpt. of Telecommunications, Technical University of Ostrava and he is also a researcher with Dpt. of Multimedia in Association of Czech universities and Czech Academy of Sciences - CESNET, Zikova 4, 160 00 Prague 6, Czech Republic (corresponding author provides phone: +420-603565965; e-mail: voznak@ ieee.org).

F. Rezac is an Assistant Professor with Dpt. of Telecommunications, Technical University of Ostrava and he is also a researcher with Dpt. of Multimedia in CESNET, Czech Republic (e-mail: filip@cesnet.cz).

J. Rozhon is a PhD. student with Dpt. of Telecommunications, Technical University of Ostrava and he is also a researcher with Dpt. of Multimedia in CESNET, Czech Republic (e-mail: rozhon@cesnet.cz).

II. State of The Art

In this chapter, the intrusive and non-intrusive approach is discussed. We focus especially on the latter which is an object of the proposed simplified E-model and its implementation.

A. Intrusive Methods

The intrusive methods use the original voice sample as it has entered the communication system and compare it with the degraded one as it has been outputted by this transmission chain [3], [4], [5]. The following list contains the most important intrusive algorithms:

- Perceptual Speech Quality Measurement PSQM (ITU-T P.861),
- Perceptual Analysis Measurement System PAMS (time alignment algorithm was adopted by ITU-T P.862),
- Perceptual Evaluation of Speech Quality PESQ (P.862),
- and Perceptual Objective Listening Quality Assessment P.OLQA (ITU-T P.862).

Among these, PESQ is currently the most commonly applied algorithm [4]. The basic philosophy of the PESQ approach is depicted on Fig.1. As stated above, the principle of this intrusive test is the comparison of original and degraded signals, their mathematical analysis and interpretation in the cognitive model. The model begins by level aligning both signals to a standard listening level. They are filtered (using an FFT) with an input filter to model a standard telephone handset. The signals are aligned in time and then processed through an auditory transform similar to that of PSQM. The transformation also involves equalizing for linear filtering in the system and for gain variation. Two distortion parameters are extracted from the disturbance (the difference between the transforms of the signals) and are aggregated in frequency and time and mapped to a prediction of subjective mean opinion score (MOS).

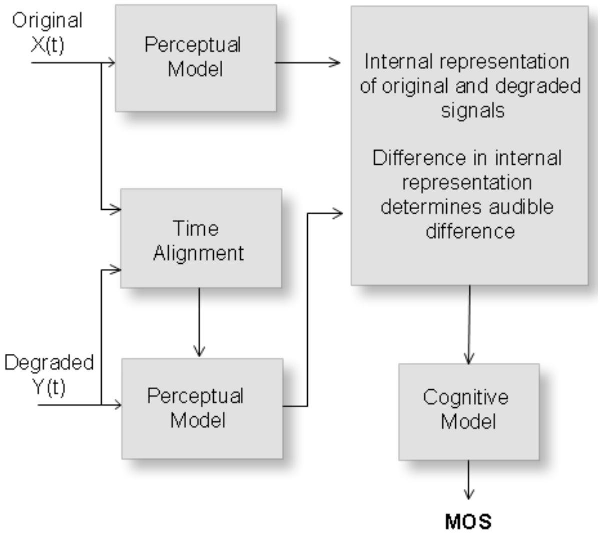


Fig. 1 The basic philosophy used in PESQ

B. Non-Intrusive Methods

Contrary to intrusive methods which require both the output (degraded) sample and the original sample, non-intrusive methods do not require the original sample [6], [7]. This is why they are more suitable to be applied in real time. Yet, since the original sample is not included, these methods frequently contain far more complex computation models. Examples of such measurement types frequently INMD (in-service non-intrusive measurement device) that has access to transmission channels and can collate objective information about calls in progress without disrupting them. The method defined by ITU-T recommendation P.562 or a more recent computation method E-model defined by ITU-T recommendation G.107 are examples of such measurements [8].

Intrusive methods are very precise but their application in real-time measurement is unsuitable because they require sending a calibrated sample and both endpoints of the examined communication. Non-intrusive approaches investigate the receiving signal. Two basic principles exist: a source-based approach and a priori-based.

The source-based approach, is based on knowledge of various types of impairments, i.e. a set of all impairments gained by comparison of original and degraded signal characteristics. The PLP (Perceptual-linear Prediction) model is a representative of this approach. PLP compares the perceptual vectors extracted from examined samples with the untainted vectors gained from original samples. Later the PLP model was modified and the computation was accelerated, nevertheless this model is not suitable for implementation in practice as its accuracy strongly depends on the quality of the database with patterns.

As for the latter approach, we would like to mention the pioneer work of Zoran and Plakal [9]. They applied artificial neural networks (ANN) to determine statistical ties between a subjective opinion and a characteristic deformation in the received sample. They also investigated spectrograms (a spectrogram is defined as a two-dimensional graphical representation of a spectrum varying in time) and they were able to establish typical uniform aspects of speech in spectrograms.

The important method was standardized in recommendation ITU-T P.562 (INMD) and in ITU-T G.107, so-called E-model. INMD measurement (In-service Non-intrusive Measurement Devices) is applied primarily to measure voice-grade parameters such as speech, noise and echo. The output from the model is a prediction of customer opinion Y_C^B (1).

$$Y_C^B = 1 + (E^B \cdot Y_{Cpre-echo}^B) \tag{1}$$

E^B is an echo and a delay multiplier, its value is between zero and one, to modify the pre-echo opinion score to take account of echo and delay impairments. $Y_{Cpre-echo}^B$ is the calculated pre-echo opinion score, on a zero-to-four scale, which takes into account effects of noise and loss. The addition of one converts Y_C^B to a one-to-five scale. All intermediate opinion score values are based on a zero-to-four scale for ease of calculation. It is possible to calculate a rating R in relation (2) using INMD measurements for a connection which is translated into a customer opinion of E-model [8].

$$R = R_0 - I_{OLR} - I_{DD} - I_{e-eff} - I_{DTE} \tag{2}$$

R_0 is the signal-to-noise ratio at a 0 dB reference point. In the equations provided (2), the 0 dB reference point is at the 2-wire input to the telephone receiving system at the near end of the connection. I_{OLR} represents the impairment term for the overall loudness rating, I_{DD} the impairment term for the absolute one-way delay and I_{e-eff} is the impairment term for the low bit-rate coding under random packet loss conditions. Last parameter I_{DTE} represents the impairment term for the delayed talker echo.

III. E-MODEL

The E-model is a computation model which takes into account all links between transmission parameters. Its output is a scalar labeled R which is a function of total expected call quality and is expressed by the following equation (3):

$$R = R_o - I_s - I_d - I_{e-eff} + A \tag{3}$$

R_0 represents the signal-to-noise ratio and includes all types of noise. The factor I_S comprises all possible impairments combinations that appear more or less simultaneously with a useful voice signal. Factor I_d represents all impairments which are caused by different combinations of delays [10], [11]. This impairment factor is expressed by the relation (4) where it is subdivided into the three factors.

$$I_d = I_{dte} + I_{dle} + I_{dd} \tag{4}$$

The factor *Idte* gives an estimate for the impairments due to talker echo and the factor *Idle* represents impairments due to listener echo. Factor *Idd* represents the impairment caused by too-long absolute delay *Ta* which occurs even with perfect echo cancelling. For Ta ≤ 100 ms we can assume *Idd = 0* because a negligible influence appears in the R-factor but with the delay increasing the overall R-factor is affected, see Fig. 2.

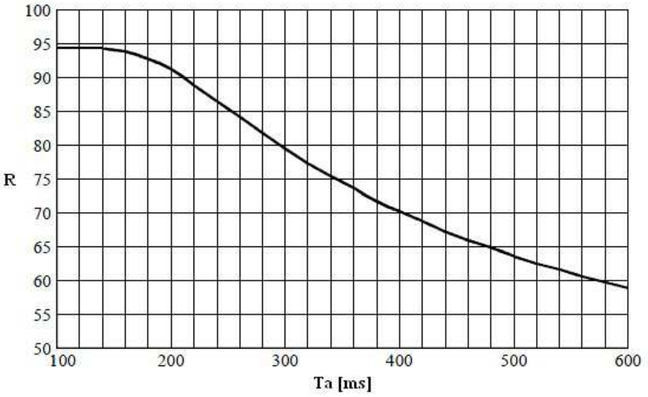


Fig. 2 Dependency of R-factor on absolute delay

$I_{e\text{-}eff}$ comprises impairments caused by using a particular voice codec, occurrence of packet loss and its resistance against losses. Specific impairment factor values for codec operation under random packet-loss have formerly been treated using tabulated, packet-loss dependent *Ie* values. Now, the packet-loss robustness Factor *Bpl* is defined as a codec-specific value. The packet-loss dependent effective equipment impairment factor *Ie-eff* is derived using the codec-specific value for the equipment impairment factor at zero packet-loss *Ie* and the packet-loss robustness factor *Bpl*, both listed in Appendix I of ITU-T G.113 for several codecs [12]. With the packet-loss probability *Ppl*, *Ie-eff* is calculated using the equation (5).

$$I_{e-eff} = I_e + (95 - I_e) \cdot \frac{P_{pl}}{\frac{P_{pl}}{BurstR} + B_{pl}} \tag{5}$$

BurstR is the so-called burst ratio, defined as ratio between “Average length of observed bursts in an arrival sequence“ and “Average length of bursts expected for the network under random loss“.

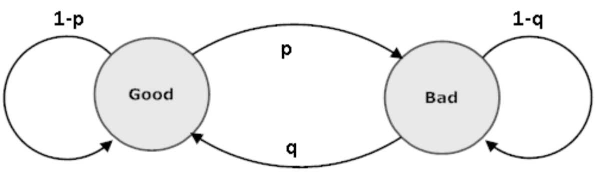


Fig. 3 Gilbert-Elliot model

Where the packet loss is random, BurstR=1 and in case of packet loss burstiness, BurstR>1. As packet loss distributions correspond to the Gilbert-Elliot model with transition probabilities p between the found and a loss state, and q between the loss and the found state, the burst ratio can be calculated as *p/(p+q)*. Fig. 3 illustrates such a situation, the packet loss Ppl is expressed by relation (6) and BurstR can be calculated as follows (7):

$$P_{pl} = 100 \cdot \frac{p}{p+q} \tag{6}$$

$$BurstR = \frac{1 - P_{pl}}{q} \cdot 100 \tag{7}$$

As can be seen from equation (5), the effective equipment impairment factor in case of Ppl = 0 (no packet-loss) equals Ie value defined in Appendix I of ITU-T G.113 [12], see Tab.1.

Tab. 1 Tables of Ie values assigned to individual codecs.

Codec Type	Reference	Operating Rate [kbit/s]	I_e value
PCM	***G.711***	***64***	***0***
ADPCM	***G.726, G.727***	***40***	***2***
	G.721, G.726, G.727	***32***	***7***
	G.726, G.727	***24***	***25***
	G.726, G.727	***16***	***50***
LD-CELP	***G.728***	***16***	***7***
		12.8	***20***
CS-ACELP	***G.729***	***8***	***10***
	G.729-A + VAD	***8***	***11***
VSELP	***IS-54***	***8***	***20***
ACELP	***IS-641***	***7.4***	***10***
QCELP	***IS-96a***	***8***	***21***
RCELP	***IS-127***	***8***	***6***
VSELP	***Japanese PDC***	***6.7***	***24***
RPE-LTP	***GSM 06.10, full-rate***	***13***	***20***
VSELP	***GSM 06.20, half-rate***	***5.6***	***23***
ACELP	***GSM 06.60, enhanced full-rate***	***12.2***	***5***
ACELP	***G.723.1***	***5.3***	***19***
MP-MLQ	***G.723.1***	***6.3***	***15***

Finally, parameter slightly adjusts the final quality depending on user’s concentration. The value of conventional (wire-bound) communication system is A=0, mobility by cellular networks in a building A=5, mobility in a geographical area or moving in a vehicle A=10 and access to hard-to-reach locations, e.g. via multi-hop satellite connections A=20. For all input parameters used in the E-model’s algorithm, the default values are listed in recommendation ITU-T G.107. While planning the calculation, it is strongly recommended to use these default values for all parameters which do not vary.

IV. Technology and Methodology Used

This chapter deals with the application of the computational E-model, simplified for the purpose of implementation.`

A. *Simplified E-Model*

The simplified E-model takes into account only effects from codec, packet loss (random packet loss) and end-to-end delay. Fig. 4 illustrates the situation which corresponds to relation (9).

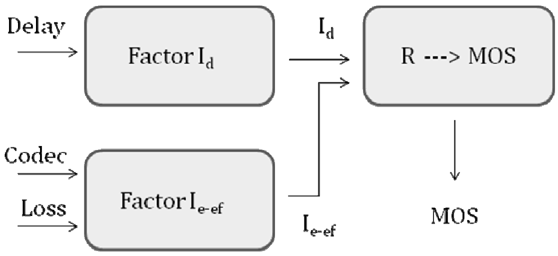


Fig. 4 E-model in simplified version.

As for the codec, it is simply identified at the receiving side. We applied a linear regression to results gained in AT&T laboratories [13] and derived relation (8) which provides accurate results, with regression quality r=0.99 ranging from 0 to 400 ms.

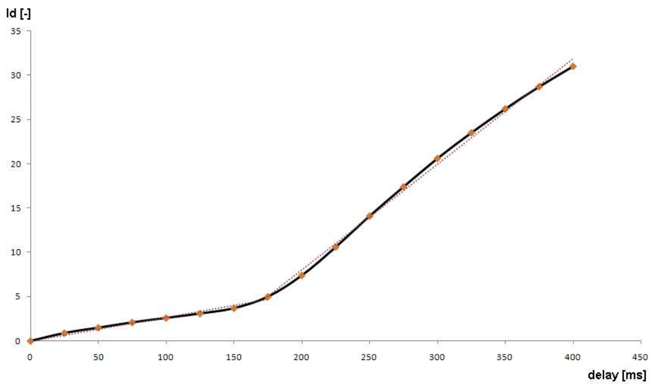


Fig. 5 Delay Impairment factor Id dependency on delay.

$$I_d = \begin{cases} 0.0267 \cdot T & T < 175\text{ms} \\ 0.1194 \cdot T - 15.876 & 175\text{ms} \le T \le 400\text{ms} \end{cases} \quad (8)$$

Parameters R_0, I_S and A are replaced by constants, with their values stated in recommendation ITU-T G.107. The original relation (3) has been modified as follows (9):

$$R = 94.7688 - 1.4136 - I_d - I_{e-eff} + 0 \quad (9)$$

Parameter I_{e-eff} is computed in relation (5). Where the packet loss distribution is unknown, the value of the packet loss is assumed as random and BurstR = 1 and it results in the following simplification. Parameter Ie is fully taken over from recommendation ITU-T G.113 where its values for the most used codecs are listed, see Tab. 2. (PLC stands for a Packet Loss Concealment).

Tab. 2 Relation between Bpl and Ie for several codecs.

Codec	Packet Size	PLC type	I_e value	B_{pl} value
G.723.1 + VAD	30 ms	Native	15	16.1
G.729A + VAD	20 ms (2 frames)	Native	11	19.0
GSM - EFR	20 ms	Native	5	10.0
G.711	10 ms	None	0	4.3
G.711	10 ms	Appendix I of [ITU-T G.711]	0	25.1

Finally, the computed R-factor is converted to MOS value. For this purpose, relation (10) was adopted [8]. MOS values > 100 can be achieved only provided a wide-band codec is used.

$$\begin{aligned} &MOS = 1 \ \text{ for } \ R < 6.5 \\ &MOS = 1 + 0.035 \cdot R + R \cdot (R - 60) \cdot (100 - R) \cdot 7 \cdot 10^{-6} \\ &\qquad\qquad \text{for } \ 6.5 \le R \le 100 \\ &MOS = 4.5 \ \text{ for } \ R > 100 \end{aligned} \quad (10)$$

Relation between MOS and R-factor is depicted in Fig. 6.

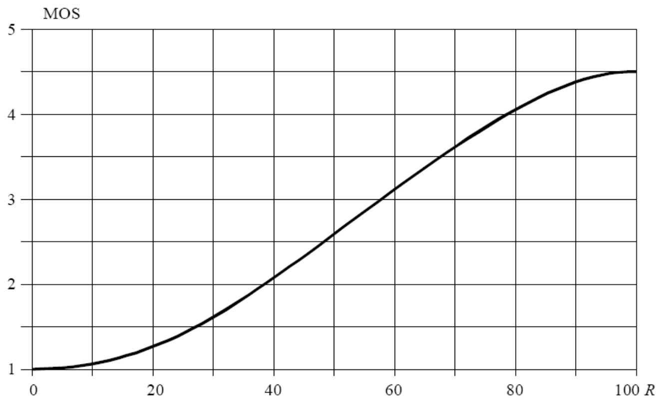


Fig. 6 Dependency between MOS and R-factor.

B. *Implementation*

Our tool was implemented mainly in Java, the code communicates with SQLite3 database via API connector SQLiteJDBC. Together with several scripts written in shell and php, they create our platform. The proposed approach was implemented in the BESIP system (Bright Embedded Solution for IP telephony) [14]. The structure of our VQM (Voice Quality Module) is depicted in Fig. 7.

The system itself consists of three logical components: being a web interface that serves the administrators (Web GUI), part of the script (Scripts) that controls obtaining information necessary to compute the speech quality in the simplified E-model and the Quality Monitor, which contains the logic for the calculation itself and processes data obtained through scripts. Results are stored in a SQLite3 database. The developed application offers the comfort of management in a web application while the developed interface aggregates required functions.

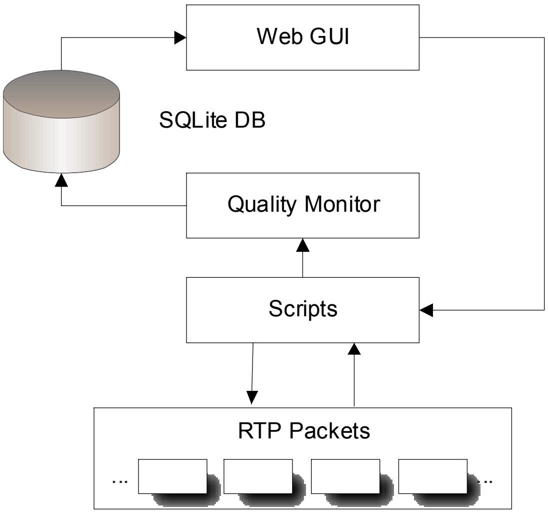


Fig. 7 Overview of VQM logical structure.

The structure of the presented data is as follows: Time, Source IP, Destination IP, MOS, R-factor and Codec. Scripts are launched through the web interface of the monitoring tool enabling the monitoring itself, the Web GUI is depicted in Fig. 8.

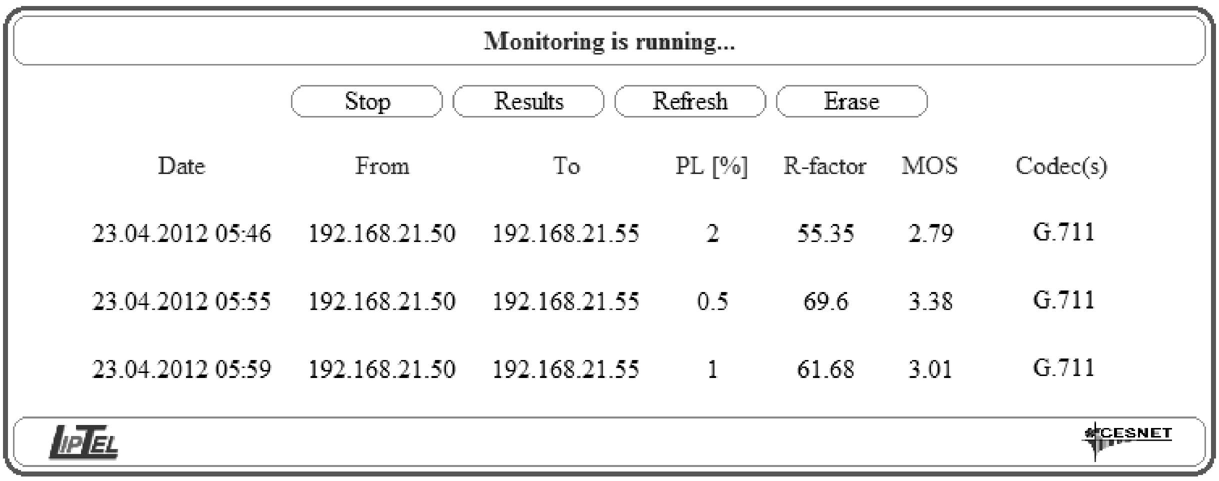


Fig. 8: Web GUI of speech quality monitoring system.

The management is implemented using the four buttons located below the indicator status of the application.

Start/Stop – Depending on the current state, (on/off) the first button is either Start or Stop. As its name suggests, it serves to activate/deactivate the monitoring process. By clicking the Start button, the PHP script _start.php is launched at the server side using the Java script and Ajax which launches the _start.sh shell script in the background. It activates tshark. It is terminated every 900 seconds and data obtained from the RTP filter are written into the sterams.log file. At this stage, the recalculation is not performed after each step. The script automatically computes the parameters and clears streams.log after three predefined rotations.. By clicking the Stop button, the _stop.php script is launched by using the Java script and Ajax at the server side. The _stop.sh shell script terminates any tshark and sh_start.sh processes by means of the kill command.

Results – This button is used to initialize the speech quality computing manually. The application automatically calculates the results after 3 finished cycles of tshark, i.e. after 45 minutes. Clicking this button we can calculate and display results immediately. The button reloads the page. Where there is a large number of calls after the last calculation, you need to refresh the page several times to display all results correctly. By clicking the Results button, the _perform.php script is launched using the Java script and Ajax at the server side which runs the _perform.sh shell script. Where the background monitoring is currently running, the php script first launches _stop.sh, then performs the calculation (_perform.sh) and finally resumes the monitoring using _start.sh. Where the monitoring is turned off, only the part with the the_perform.sh script is performed. The _perform.sh shell script contains the start of Java application QualityParser that has been written for this purpose and then deletes the contents of the file that stores information from tshark (streams.log) as the data had been processed.

Refresh – The Refresh button merely refreshes the page. It performs exactly the same function as the F5 key of the recovery in the web browser.

Erase – Deletes all records in the database. Pressing this button does not refresh the page automatically and the page must be refreshed manually (by using the Refresh button). The function is applied to erase all records from the SQL database. In the main (lower) part of the GUI, the results obtained are displayed in the pre-defined format.

V. Verification and Results

In order to verify the suitability of the method applied in the Voice Monitoring Module, a comparison with the objective intrusive PESQ method was carried out.

In case of using different codecs at users' sides, our application detects such situation and replaces computation of $I_{e\text{-}ef}$ parameters in equation (11), where we take into account the mean of both $I_{e\text{-}ef}$.

$$I_{e-ef} = \frac{I_{e-ef\,1} + I_{e-ef\,2}}{2} \tag{11}$$

Following sub-chapters contain a description of testbed used for verification of the proposed method and the achieved results.

A. Testbed Used for Verification

The packet loss was emulated in an open source tool *netem* [15]. This tool enables to emulate random packet loss in linux command line:

```
tc qdisc add dev eth0 root netem loss 0.5%
```

The testing calls were generated by sipp tool and we have gained a lot of experience in development applications based on sipp such as SPITFILE which works as a VoIP spam generator [16] or SIPB [17] providing a performance evaluation of SIP infrastructure. We prepared a testing scenario for sipp in XML where the structure of SIP communication is defined, it includes all steps from a registration, through an initiation of calls with negotiation of media up to a termination of established session. The situation is depicted in Fig. 9.

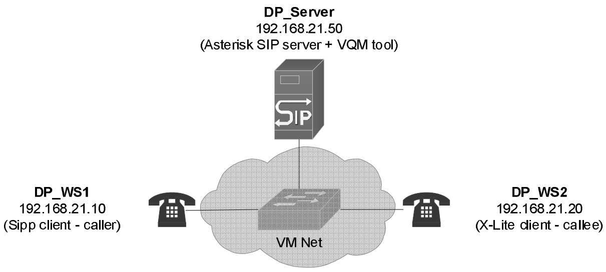


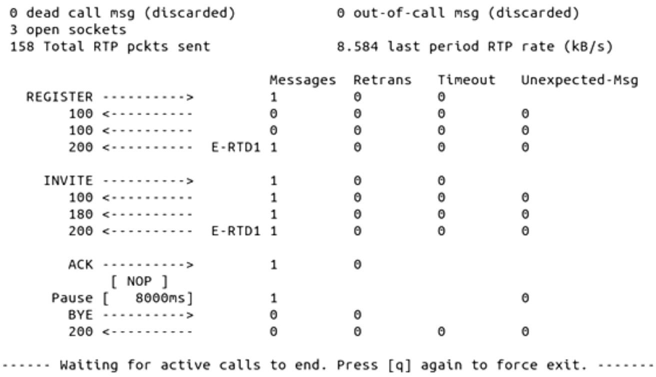
Fig. 9 Testbed for verification of applied method.

The test is launched by following command:

```
sipp -aa -r 1 -t un -sf proxy.xml
192.168.21.50 -s 101 -i 192.168.21.10 -m 1
```

The first parameter *–aa* activates an automatic sending 200 OK responses to the received SIP request INFO, UPDATE and NOTIFY. The second parameter *-r 1* indicates a rate of the generated calls per second and the used one call per second is enough for our purpose. Next switch *-t un* will set type of transport protocol UDP and a new socket will be created for every new call. The parameter "-sf proxy.xml" determines a path to the file with XML scenario. The IP address *192.168.21.50* belongs to SIP server and *192.168.21.10* to the output interface used for generation of calls. The account which is used at SIP server for registration and a call generation is specified by parameter *-s 101*, the last parameter *-m 1* indicates a termination after single call.

We could observe the individual steps of the defined scenario in console of sipp tool, as is depicted in Fig. 10.



```
0 dead call msg (discarded)            0 out-of-call msg (discarded)
3 open sockets
158 Total RTP pckts sent               8.584 last period RTP rate (kB/s)

                                 Messages  Retrans   Timeout   Unexpected-Msg
  REGISTER ---------->           1         0         0
       100 <----------           0         0         0         0
       100 <----------           0         0         0         0
       200 <----------  E-RTD1   1         0         0         0

    INVITE ---------->           1         0         0
       100 <----------           1         0         0         0
       180 <----------           1         0         0         0
       200 <----------  E-RTD1   1         0         0         0

       ACK ---------->           1         0
            [ NOP ]
     Pause [   8000ms]           1                             0
       BYE ---------->           0         0
       200 <----------           0         0         0         0

------ Waiting for active calls to end. Press [q] again to force exit. ------
```

Fig. 10 A console of running sipp application.

The structure of the prepared scenario in XML listed below contains only part this scenario valid for SIP request INVITE.

```
<send retrans="500">
<![CDATA[
   INVITE sip:102@[remote_ip] SIP/2.0
   Via:SIP/2.0/[transport][local_ip]:[local_por
   t];branch=[branch]
  From:[service]<sip:[service]@[remote_ip]>;ta
  g=[call_number]
  To: 102 <sip:102@[remote_ip]>
  Call-ID: d///[call_id]
  CSeq: 3 INVITE
  User-Agent: Grandstream GXP2000 1.1.6.37
  Contact:<sip:[service]@[local_ip]:[local_por
  t];transport=[transport]>
  Max-Forwards: 70
  Allow:INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPT
  IONS,INFO,SUBSCRIBE,UPDATE,PRACK,MESSAGE
  Supported: replaces, timer, path
  Subject: Performance Test
  Content-Type: application/sdp
  Content-Length: [len]

  v=0
  o=[service]  8000  8001  IN  IP[local_ip_type]
  [local_ip]
  s=SIP Call
  c=IN IP[media_ip_type] [media_ip]
  t=0 0
  m=audio [auto_media_port] RTP/AVP 0
  a=rtpmap:0 PCMU/8000
  a=ptime:20
  a=sendrecv
]]>
</send>
```

Multimedia data are passed on to the scenario of sipp tool in a file with a defined format. The format is *pcap*, hence the original calibrated *wav* sample has to be converted to *pcap*.

Fro this purpose, we adopted *wav2rtp* application where conversion is executed by following command in Linux CLI:

```
wav2rtp –f input.wav –t output.pcap –c PCMU
```

The last parameter *-c PCMU* defines a codec which will be used for conversion and we prepared speech samples for requested set of codecs.

As we have mentioned, the comparison of MOS computed in our VQM application with the objective intrusive PESQ method was carried out. PESQ on its own is a listening model, so PESQ quality scores do not normally take account of the conversational factors: level, talker echo and sidetone. Hence the delay was not accounted in experiments. A calibrated sample was transmitted and the range of measurements was repeated up to 10% of packet loss with step 0.5. Only several codecs included in BESIP [14] were examined: G.711 A-law, G.711 µ-law, G.729 and GSM-EFR. The PESQ algorithm was adopted from ITU-T P.862, the source code in C++ is a part of the recommendation. We compiled this code in linux command line and the we were able to determine MOS by PESQ method where we needed both the original and the degraded samples. The structure of command is following:

```
./pesq +8000 reference.wav degraded.wav
```

The original sample was stored at caller side and used by sipp tool, whereas the degraded sample was captured by network analyzer at callee side. For this purpose, we exploit application Wireshark which enables to store RTP streams and export their payload to the individual wav files. An example of the gained MOS in PESQ is depicted in Fig. 11.

```
Reading reference file test_r.wav...done.
Reading degraded file test_d.wav...done.
 Level normalization...
 IRS filtering...
 Variable delay compensation...
 Acoustic model processing...

Prediction : PESQ_MOS = 2.991
```

Fig. 11 An example of gained MOS in PESQ.

B. *Comparison of MOS Gained from PESQ and MOS Computed in VQM based on E-Model*

MOS was computed for every measurement and measurements were repeated five times for each codec and each loss value to eliminate possible statistical variances and errors. In order to evaluate the variance between MOS values gained from PESQ and the E-Model, the correlation function was applied (12) where individual parameters are given in the following relations (13) and (14).

$$r = \frac{S_{XY}}{\sqrt{S_X^2 \cdot S_Y^2}} \text{ for } S_X, S_Y \neq 0 \qquad (12)$$

Relations (13), (14) represent a variance of values in individual methods and values $\overline{X}$, $\overline{Y}$ represent an arithmetic mean of individual values gained from the E-Model and PESQ.

$$S_X^2 = \frac{1}{n-1} \cdot \sum_{i=1}^{n} \left(X_i - \overline{X} \right)^2 \qquad (13)$$

$$S_Y^2 = \frac{1}{n-1} \cdot \sum_{i=1}^{n} \left(Y_i - \overline{Y} \right)^2 \qquad (14)$$

The deviations between values are minimal and conformity is higher than 98% for all codecs compared (see Table 5), a graphical representation for GSM EFR codec is depicted in Fig. 12 and listed in Tab. 4.

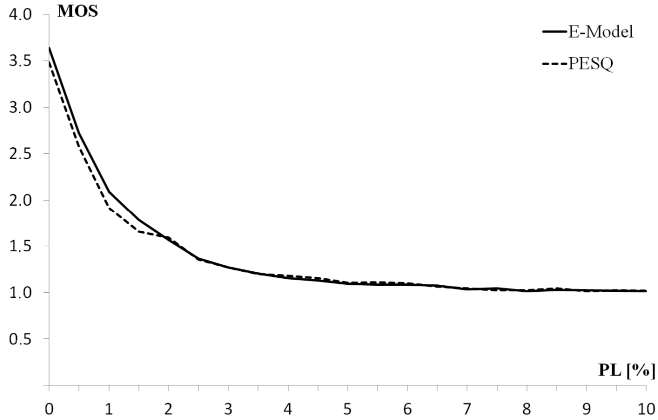


Fig. 12 Comparison of MOS computed in E-Model and gained from PESQ for GSM EFR codec.

The applied method proved to be efficient and suitable for the purpose of speech quality. We have take into account that PESQ is the intrusive method and is considered as very precise for narrow-band codecs whereas simplified E-model needs only information about original codec extracted from payload type of RTP packet, packet loss and delay.

Table 4. MOS values computed in simplified E-Model and gained from PESQ for GSM EFR codec.

Packet Loss [%]	Computed MOS in E-model	Gained MOS from PESQ
0.00	3.73	3.48
0.50	2.72	2.58
1.00	2.09	1.91
1.50	1.78	1.66
2.00	1.57	1.59
2.50	1.37	1.35

3.00	1.27	1.27
3.50	1.20	1.20
4.00	1.16	1.18
4.50	1.13	1.16
5.00	1.10	1.11
5.50	1.09	1.11
6.00	1.09	1.10
6.50	1.08	1.06
7.00	1.03	1.04
7.50	1.04	1.02
8.00	1.02	1.03
8.50	1.03	1.04
9.00	1.02	1.01
9.50	1.02	1.03
10.00	1.01	1.02

Table 5. Correlation between PESQ and the implemented simplified E-Model for codecs used in the BESIP project.

Codec	Correlation [%]
G.711 A-law	98,95
G.711 μ-law	98,61
GSM EFR	99,87
G.729	99,88

VI. Conclusion

The paper deals with the proposed simplified E-model and its implementation in the Monitoring module of the BESIP project. In order to verify results, PESQ was applied in accordance with ITU recommendation P.862. Within the observed range of packet losses, the proposed model achieves a decent correlation to PESQ. The performed experiment confirmed that the proposed simplified E-model corresponds with the results obtained from PESQ for selected codecs with minimal deviations and conformity higher than 98%.

The second contribution of the paper lies in the design and the practical implementation of the Monitoring module in BESIP which enables assessing the speech quality. Further improvements are possible in the support of more codecs, acceleration and optimization of the implemented algorithm. Consequently, a log agent for the selected monitoring system such as Nagios or Zabbix could be developed in near future.

Acknowledgment

This research has been supported by the Ministry of Education of the Czech Republic within the project LM2010005.

References

[1] J. Karam, "A Comprehensive Approach for Speech Related Multimedia Applications," *WSEAS Transactions on Signal Processing*, Issue 1, Volume 6, January 2010. pp. 12-21.

[2] ITU-T, Methods for subjective determination of transmission quality, ITU-T P.800, Geneva, 1996.

[3] L. Estrada, D. Torres, H. Toral, "Analytical Description of a Parameter-based Optimization of the Quality of Service for VoIP Communications," *WSEAS Transactions on communications*, Issue 9, Volume 8, September 2009, pp. 1042-1052.

[4] A. Rix, M. Hollier, A. Hekstra, J. Beerends, Perceptual evaluation of speech quality (PESQ): The new ITU standard for end-to-end speech quality assessment, *AES: Journal of the Audio Engineering Society*, Volume 50, Issue 10, 2002, pp. 755-764.

[5] R. Blatnik, G. Kandus, T. Javornik, "VoIP/VoWLAN system performance evaluation with low cost experimental test-bed," *WSEAS Transactions on Communications*, Volume 6, Issue 1, January 2007, Pages 209-216.

[6] M. Voznak, "E-model modification for case of cascade codecs arrangement," *International Journal of Mathematical Models and Methods in Applied Sciences*, Volume 5, Issue 8, 2011, pp. 1439-1447.

[7] M. Voznak, M. Tomes, Z. Vaclavikova, M. Halas, "E-model improvement for speech quality evaluation including codecs tandeming," In Proc. International Conference on Data Networks, Communications, 2010, pp. 119-124.

[8] ITU-T, The E-model: A computational model for use in transmission planning, ITU-T G.107, Geneva, 2009.

[9] J. Palakal, M. Zoran, "Feature extraction form speech spectrograms using multi-layered network models," In Proc. IEEE International Workshop on Tools for Artificial Intelligence, Architectures, Languages and Algorithms, 1989, pp. 224–230.

[10] M. Voznak, A. Kovac, M. Halas, "Effective packet loss estimation on VoIP jitter buffer," *Lecture Notes in Computer Science*, Volume 7291 LNCS, 2012, pp. 157-162.

[11] B. Kyrbashov, I. Baronak, I. Kovacik, V. Janata, "Evaluation and investigation of the delay in voip networks," *Radioengineering*, Volume 20, Issue 2, June 2011, pp. 540-547.

[12] ITU-T, Transmission impairments due to speech processing, ITU-T G.113, Geneva, 2007.

[13] G. Cole, H. Rosenbluth, Voice over IP performance monitoring, *ACM SIGCOMM Computer Communication*, New York, 2001.

[14] L. Macura, M. Voznak, K. Tomala, J. Slachta, "Embedded multiplatform SIP server solution," In Proc. 35th International Conference on Telecommunications and Signal Processing, TSP 2012, pp. 263-266.

[15] A. Jurgelionis, J. Laulajainen, M. Hirvonen, A. Wang, "An Empirical Study of NetEm Network Emulation Functionalities," In Proc. 20th International Conference on Computer Communications and Networks, 2011.

[16] M. Voznak, F. Rezac, "VoIP SPAM and a defence against this type of threat," In Proc. 14th WSEAS International Conference on Communications, 2010, pp. 172-177.

[17] M. Voznak, J. Rozhon, "Methodology for SIP infrastructure performance testing," *WSEAS Transactions on Computers*, Volume 9, Issue 9, September 2010, Pages 1012-1021.

M. Voznak (born in 1971) is an Associate Professor with Dpt. of Telecommunications, Technical University of Ostrava. He is also a researcher with Dpt. of Multimedia in CESNET (association of Czech universities and Czech Academy of Sciences). He received his MSc and PhD degrees in

telecommunications, dissertation thesis "Voice traffic optimization with regard to speech quality in network with VoIP technology" from the Technical University of Ostrava, in 1995 and 2002, respectively. Topics of his research interests are Next Generation Networks, IP telephony, speech quality and network security. He was involved in several FP EU projects, he is a member of editorial boards of several journals such as "Radioengineering", "Embedded and Distributed Systems" and "Advances in Electrical and Electronic Engineering".

F. Rezac (born in 1985) is an Assistant Professor with Dpt. of Telecommunications, Technical University of Ostrava, Czech Republic. He received his MSc in 2009 in study branch "Mobile Technology" from Technical University of Ostrava and currently continues in the doctoral study. His research is focused on Voice over IP technology, Network Security and Call Quality in VoIP. He is also a researcher with Dpt. of Multimedia in CESNET and involved in important project in the area of intelligent security systems, 7 FP EU project INDECT (2009-2013) conducted by university AGH Cracow in Poland.

J. Rozhon (born in 1986) is a Ph.D student with the Dpt. of Telecommunications, Technical University of Ostrava. He is also a researcher with Dpt. of Multimedia in CESNET. His main topic of interest is the research in the field of performance testing and benchmarking of SIP network elements. He is also active in the IP telephony in general focusing on the various implementations of Asterisk PBX and its alternatives.