Non-intrusive Speech Quality Assessment in Simplified E-Model

MIROSLAV VOZNAK
Department of Telecommunications
VSB-Technical University of Ostrava
17. listopadu 15/2172, 708 33 Ostrava Poruba
CZECH REPUBLIC
voznak@ieee.org http://voznak.eu

Abstract: - The E-model brings a modern approach to the computation of estimated quality, allowing for easy implementation. One of its advantages is that it can be applied in real time. The method is based on a mathematical computation model evaluating transmission path impairments influencing speech signal, especially delays and packet losses. These parameters, common in an IP network, can affect speech quality dramatically. The paper deals with a proposal for a simplified E-model and its practical implementation into the Monitoring module. In order to enable the comparison of results achieved, PESQ (Perceptual Evaluation of Speech Quality) was applied in accordance with ITU-T recommendation P.862, nowadays regarded as the standard for determining MOS, the measurements confirmed the applicability of the proposed method and its suitability for speech quality assessment.

Key-Words: - E-model, MOS, R-factor, PESQ, Speech quality, BESIP

1 Introduction

Methodologies evaluating speech quality can be sub-divided into two groups according to the approach applied - conversational and listening. Conversational tests are based on mutual interactive communication between two subjects through the whole transmission chain of the tested communication system. These tests provide the most realistic testing environment but they are very time consuming. Listening tests do not provide such plausibility as conversational tests but they are recommended more frequently [1]. According to the method of assessment, speech quality evaluation methodologies can be subdivided into subjective methods and objective methods. To evaluate speech quality, MOS (Mean Opinion Score) scale as defined by the ITU-T recommendation P.800 is applied [2].

2 MOS Scale

The basic scale as prescribed by the recommendation is depicted on Fig. 1. In order to avoid misunderstanding and incorrect interpretation of MOS values, ITU-T published ITU-T recommendation P.800.1 in 2003 [3]. This recommendation defines scales both for subjective and objective methods as well as for individual conversational and listening tests [4], [5].

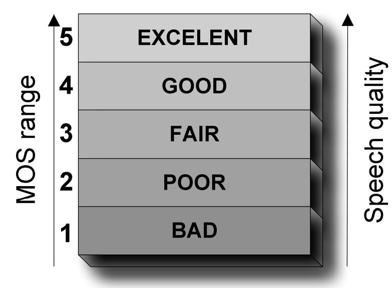

Fig. 1 MOS Scale.

2.1 Subjective Evaluation Methods

These methods are based on evaluation by human beings (listeners), i.e. subjects. During the testing, samples are played to a sufficient number of subjects, and their results are subsequently analysed statistically. Subjects can evaluate speech quality on a five-degree scale in accordance with the MOS model as defined by ITU-T.

The best known representatives of these measurements include methods such as ACR (Absolute Category Rating) or DCR (Degradation Category Rating). Major disadvantages of these methods are high requirements on time, the fact that final evaluation is influenced by listener's subjective opinion and most of all impossibility to use them for testing in real time [24].

2.2 Objective Evaluation Methods
The use of objective methods substitutes the necessity to involve humans in the testing by mathematical computational models or algorithms. Their output is again a MOS value or, depending on the algorithm applied, a different value which can be converted into a MOS value using a suitable mapping function. The aim of objective methods is to estimate, as precisely as possible, the MOS value which would be obtained by a subjective evaluation involving a sufficient number of evaluating subjects. Objective testing's exactness and efficiency is therefore a correlation of results from both subjective and objective measurements [6]. Objective methods can be sub-divided into two groups, Intrusive and Non-intrusive.

3 State of the Art
In this chapter, the intrusive and non-intrusive approach is discussed. I focus especially on the latter which is an object of the proposed simplified E-model and its implementation. The results are compared to the generally reputable intrusive approach represented by ITU-T P.862.

3.1 Intrusive Approach
The core of intrusive (also referred to as input-to-output) measurements is the comparison of the original sample and the degraded sample affected by a transmission chain [6], [7]. 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. The following list contains the most important intrusive algorithms:

- Perceptual Speech Quality Measurement PSQM
- Perceptual Analysis Measurement System PAMS
- Perceptual Evaluation of Speech Quality PESQ
- Perceptual Objective Listening Quality Assessment P.OLQA

Among these, PESQ is currently the most commonly applied algorithm [8]. It combines the advantages of PAMS (robust temporal alignment techniques) and PSQM (exact sensual perception model) and is described in ITU-T's recommendation P.862. The last algorithm mentioned, P.OLQA, also known as ITU-T P.863, is intended to be a successor of the PESQ. It strives to avoid the weaknesses of the PESQ's model and to incorporate a better wideband codec analysis in comparison with PESQ.

The basic philosophy of the PESQ approach is depicted on Fig.2. 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.

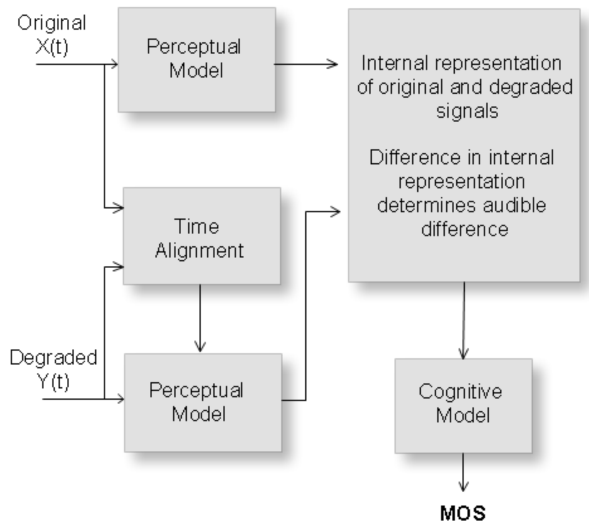

Fig. 2 The basic philosophy used in PESQ.

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 equalising 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) [9]-[11].

3.2 Non-Intrusive Approach
Contrary to intrusive methods which require both the output (degraded) sample and the original sample, non-intrusive methods do not require the original sample. 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. These data are further processed using a particular method, with a MOS value as the output [3]. 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 [12].

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. Nevertheless, we usually need to assess the speech quality in real traffic and be able to record its changes, especially degradation. Non-intrusive approaches investigate the receiving signal. Two basic principles exist:

- a source-based approach,
- and a priori-based,

The former, 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. As I have mentioned, it requires a database with the set of impairments and high computational complexity. 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, I would like to mention the pioneer work of Zoran and Plakal [13]. 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 defines 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).

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

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 generate a rating R (2) using INMD measurements for a connection which is translated into a customer opinion of E-model [10], [12]. The E-model is one of the most modern non-intrusive methods and will be described in the chapter.

$$R = R_0 - I_{OLR} - I_{DD} - I_{e-eff} - I_{DTE} \qquad (2)$$

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.

4 E-Model

The complexity of modern networks requires that individual transmission path parameters are not assessed separately but rather that all their possible combinations and their interaction are considered [14]. Partially, this can be achieved by an expert estimate based on the parameters of the transmission path, yet using a computation model is a more systemic approach. The E-model is a computation model which takes into account all the links between transmission parameters. Its output is a scalar labelled R which is a function of total expected call quality. The E-model is based on the “equipment impairment factor” method. The original structure of this model was developed by Swedish expert Nils-Olof Johannesson, member of the Voice Transmission Quality from Mouth to Ear group under ETSI. This model was further developed by the SG12 group under ITU-T and it was published in ITU-T recommendation G.107 as the E-model [8]. The structure of the connection reference model is depicted on Fig. 3.

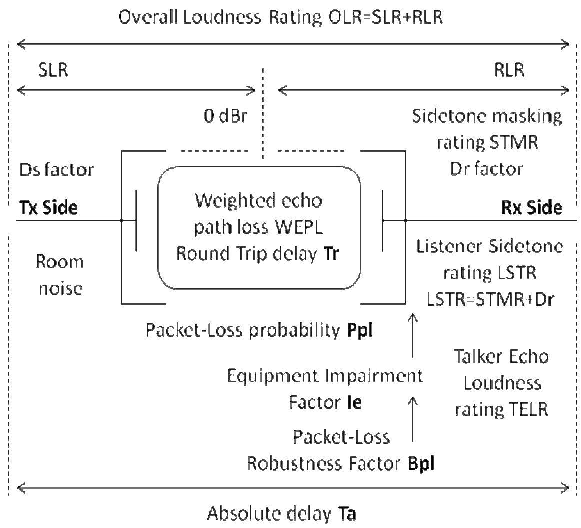

Fig. 3: Reference model for E-model computation.

The computational model consists of various mathematical operations over all parameters of the transmission system. The computation itself can be split into several elements 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, such noises caused by the device's electrical circuit and noises arisen on the wiring. 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 [14]. 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. Echo-cancellation is solved in recommendation ITU-T G.168 and can be effectively suppressed. Factor *Idd* represents the impairment caused by too-long absolute delay *Ta* which occurs even with perfect echo cancelling [8]. For Ta $\leq$ 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. This situation is depicted on Fig. 4.

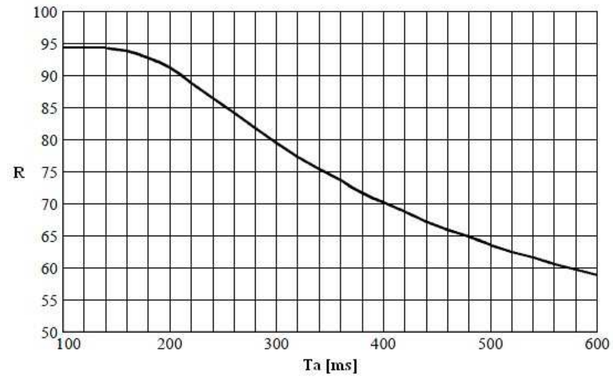

Fig. 4 Dependency of R-factor on absolute delay.

$I_{e\text{-}eff}$ 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. With the packet-loss probability *Ppl*, *Ie-eff* is calculated using the equation (5).

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

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".

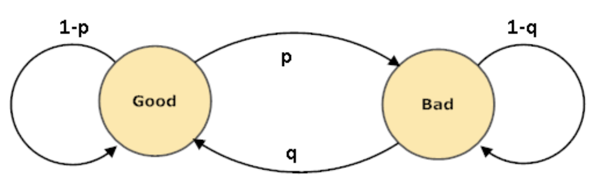

Fig. 5: Gilbert-Elliot model.

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 $p/(p+q)$ [16]. Fig. 5 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 [15].

Finally, parameter A 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. It should be noted that the above values are only provisional. The use of factor A and its selected value in a specific application is up to the planner's decision. Additional background information on the advantages of factor A can be found in Appendix II to ITU-T G.113.

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 . If all parameters are set to the default values, the calculation results in a very high quality with a rating factor of R = 93.2.

5 Technology and Methodology Used

This chapter deals with the application of the computational E-model, simplified for the purpose of implementation. First, the simplified E-model itself is explained. Next, it is implemented in JAVA. The proposed approach was implemented in the BESIP (Bright Embedded Solution for IP telephony) project, developed by the department of telecommunications at VSB-Technical University of Ostrava as an open source application [17].

5.1 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. 6 illustrates the situation which corresponds to relation (9).

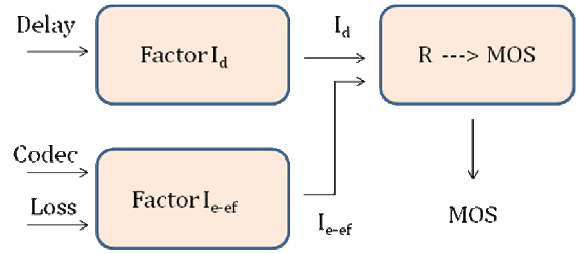

Fig. 6: E-model in simplified version.

As for the codec, it is simply identified at the receiving side. The same applies tothe delay. I applied a linear regression to results gained in AT&T laboratories [18] and derived relation (8) which provides accurate results, with regression quality r=0.99 ranging from 0 to 400 ms.

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

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):

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

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 I_e is fully taken over from recommendation ITU-T G.113 where its values for the most used codecs are listed.

Finally, the computed R-factor is converted to MOS value. For this purpose, relation (10) was adopted [12]. MOS values > 100 can be achieved only provided a wide-band codec is used.

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

5.2 Project BESIP

The BESIP (Bright Embedded Solution for IP Telephony) project has been developed by LipTel team supported in Ostrava since 2011 [17]. The primary goal is to implement a multiplatform embedded SIP communication server with a unified configuration interface. The SIP server is based on OpenWRT project core and there are Asterisk and Kamailio inside as SIP engines. The final distribution consists of four modules Core, Security, Monitoring and PBX (Private Branche Exchange) as is depicted in Fig. 7.

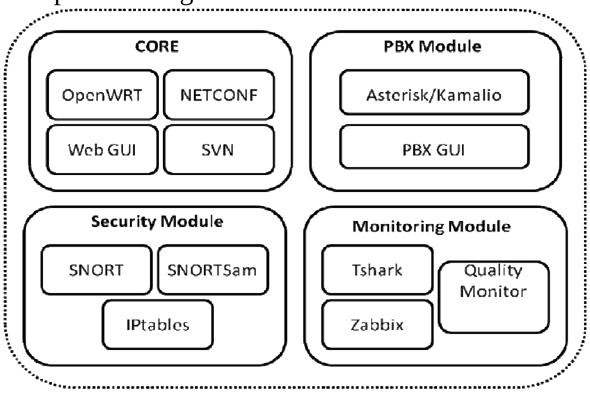

Fig. 7: BESIP modules.

The entire BESIP management and development is available at the project's homepage [19]. The project has been developed as an open source under GNU/GPL license, the binary images from nightly autobuild are freely downloadable [20].

The approach of speech quality assessment described above was implemented into the Monitoring module which forms an integral part of the official BESIP distribution [17]. As for the distribution, not only individual packages are available for download but the whole image for particular HW used for testing of pre-released distributions such as HW depicted in Fig. 8 can also be downloaded.

Fig. 8: Suitable HW platform containing x86 Intel Atom 1.6GHz, RAM 1GB/677MHz and 16GB SSD.

5.3 Implementation

The overall solution of the monitoring system consists of several different open source components and also of the part that was developed solely for this purpose to meet the defined requirements. The system structure is shown in Fig. 9. 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 structure of the application is depicted in Fig. 9.

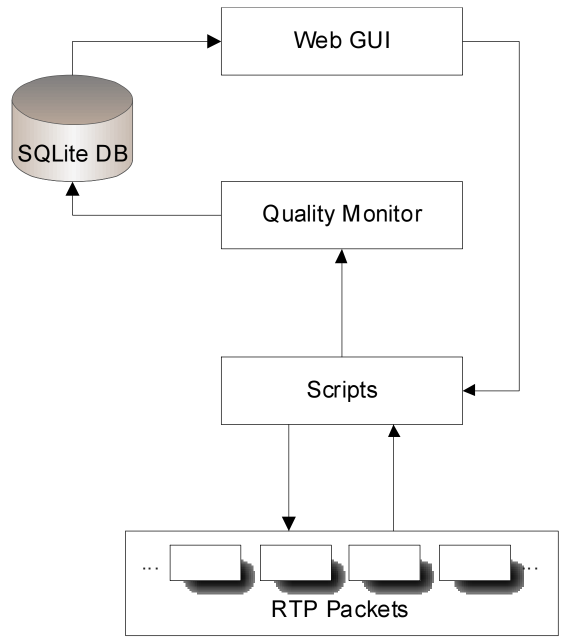

Fig. 9: Overview of VQM logical structure.

The developed application offers the comfort of management in a web application while the developed interface aggregates required functions. The web interface is the main part of user interaction with the monitoring tool. The monitoring tool is turned off in the default configuration and can be enabled any time using the BESIP's intuitive main interface. This part of the monitoring tool is also used as a means to display the measured and computed results. The structure of the presented data is as follows: Time, Source IP, Destination IP, MOS, R-factor andCodec.

The web interface is written entirely in PHP scripting language in order to enable starting or stopping the monitoring system through the OpenWRT shell as it depends on shell applications such as *tshark* (a small terminal-based network analyzer). Scripts are launched through the web interface of the monitoring tool enabling the monitoring itself. In practice, this means starting the network traffic capture with the *tshark* tool with the RTP filter activated. The usage of the RTP filter makes working with RTP streams much easier as these streams contain some important statistical data (packet loss, jitter) and other important information (source/destination IP, codec) necessary to calculate the speech quality in the E-model.

Unfortunately, the collected data are stored in a text form without a compression only after *tshark* finishes. In order to reduce the storage allocation, speech quality assessments are performed in short intervals. The period of 15 minutes (duration: 900 sec.) was chosen as a suitable one. Once the traffic data have been collated and are available, the script can start the calculation of the speech quality which is further described in the next section. After processing the obtained data are erased from the storage.

All data, as presented in chapter 5.1, are stored in a SQLite3 database. The open source tool Java API SQLiteJDBC was selected to enable communication with the database. Speech quality monitoring is controlled through a web interface that is accessible from the main crossroads management interface. Fig. 10 shows an example of the user.

The status indicator is located at the top of the GUI and indicates whether the monitoring is activated in BESIP (Monitoring is running…)or is currently turned off.

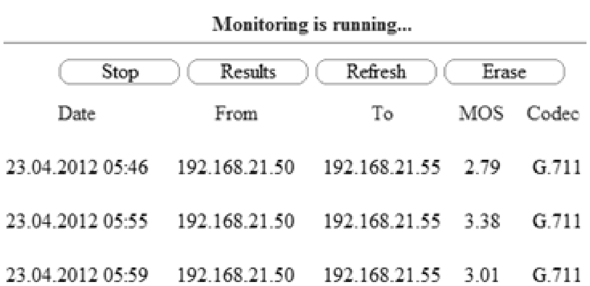

Fig. 10: Sample of web GUI of monitoring speech quality.

The management is implemented using the four buttons located below the indicator status of the application. The function state diagram is depicted in Fig. 11. **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.

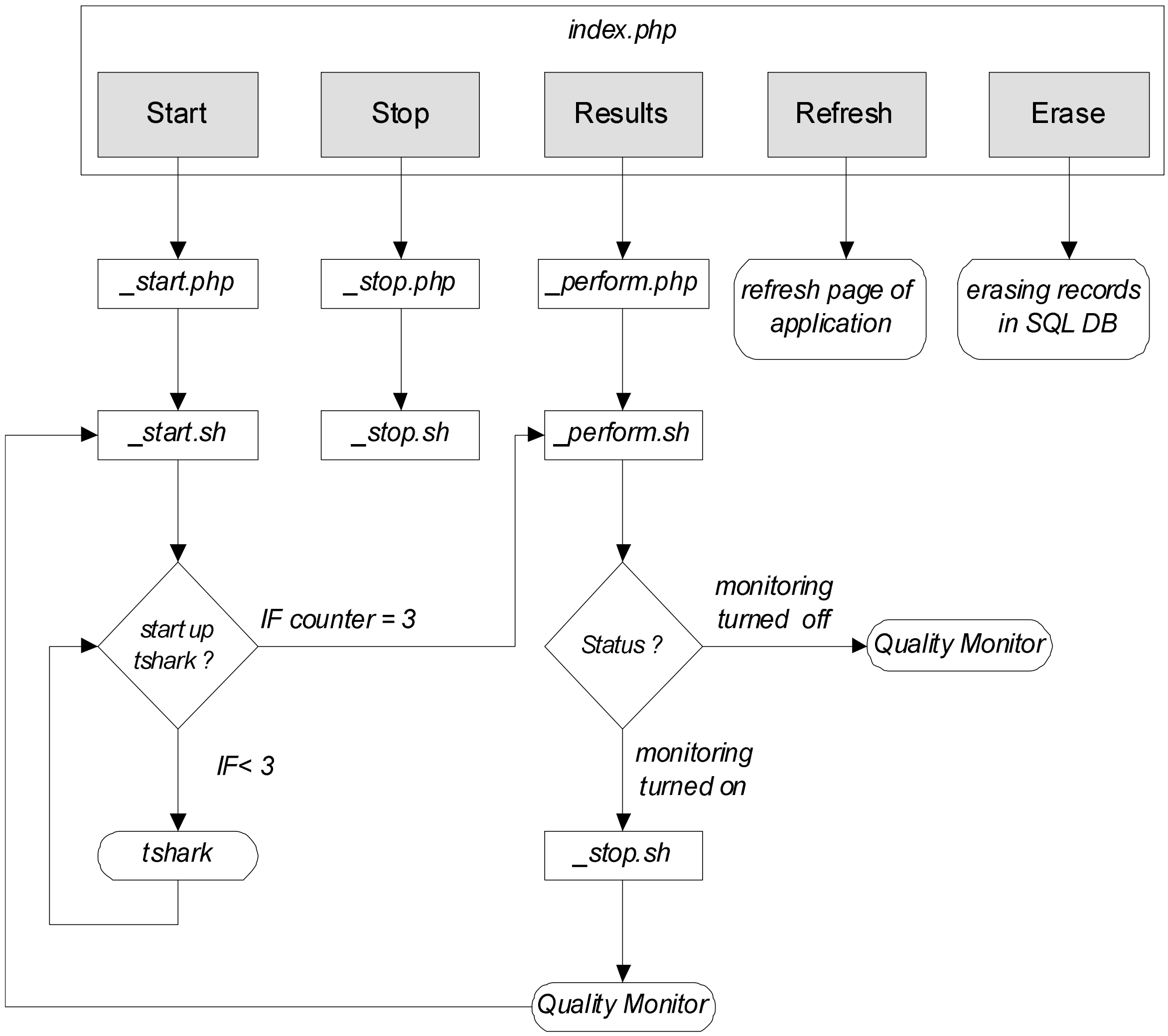

Fig. 11: The function state diagram.

6 Verification and Results

In order to verify the suitability of the method applied in the Monitoring module, a comparison with the objective intrusive PESQ method was carried out. The packet loss was emulated in an open source tool *netem*. The tool provides a Network Emulation functionality for testing protocols and the current version emulates variable delay, loss, duplication and re-ordering. *Netem* is a part of the core Linux system with kernel 2.6 or higher [21].

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 for in experiments.

The testing calls were generated by our own tool SPITFILE which had originally been developed as a VoIP spam generator [22]. A calibrated sample was sent to BESIP and subsequently analyzed in the Monitoring Module. The *netem* tool affected the speech quality and the impairment was indicated in the Monitoring module. The received degraded sample was processed by PESQ and the computed MOS value from the simplified E-Model was compared to MOS value gained from PESQ. The range of measurements was repeated up to 10% of packet loss with step 0.5. Only several codecs included in BESIP 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. Therefore, the code was downloaded from ITU-T, compiled by *gcc* (GNU Compiler Collection) in Linux. PESQ_MOS was computed for every measurement., see Fig. 12.

```
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. 12: MOS gained from PESQ.

Measurements were carried out five times for each codec and each loss value in order to eliminate possible statistical variances and errors. In order to evaluate the variance between MOS values gained from PESQ and the E-Model, the following correlation function was applied (11).

$$r = \frac{S_{XY}}{\sqrt{S_X^2 \cdot S_Y^2}} \text{ for } S_X, S_Y \neq 0 \tag{11}$$

where individual parameters are given in the following relations (12)-(16).

$$S_X^2 = \frac{1}{n-1} \cdot \sum_{i=1}^{n} \left(X_i - \overline{X} \right)^2 \tag{12}$$

$$S_Y^2 = \frac{1}{n-1} \cdot \sum_{i=1}^{n} \left(Y_i - \overline{Y} \right)^2 \tag{13}$$

Relations (12) and (13) represent a variance of values in individual methods. The covariance is expressed in (15).

$$S_{XY} = \frac{1}{n-1} \cdot \sum_{i=1}^{n} \left(X_i - \overline{X} \right) \cdot \left(Y_i - \overline{Y} \right) =$$

$$\frac{\sum_{i=1}^{n} X_i \cdot Y - n\overline{X}\overline{Y}}{\sqrt{\left(\sum_{i=1}^{n} X_i^2 - n\overline{X}^2 \right)} \cdot \sqrt{\left(\sum_{i=1}^{n} Y_i^2 - n\overline{Y}^2 \right)}} \tag{14}$$

Values $\overline{X}$ and $\overline{Y}$ in relations (15) and (16) represent an arithmetic mean of individual values gained from the E-Model and PESQ.

$$\overline{X} = \frac{1}{n} \cdot \sum_{i=1}^{n} X_i \tag{15}$$

$$\overline{Y} = \frac{1}{n} \cdot \sum_{i=1}^{n} Y_i \tag{16}$$

The deviations between values are minimal and conformity is higher than 98% for all codecs compared (see Table 1).

Table 1. Correlation between PESQ and the implemented simplified E-Model

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

The applied method proved to be efficient and suitable for the purpose of speech quality assessment.

7 Conclusion

The E-model brings a modern approach to the computation of estimated quality, allowing for easy implementation. One of its advantages is that it can be applied in real time. The method is based on a mathematical computation model and can be applied as early as the planning stage a new communication system. The E-model is classified as objective non-intrusive method and is applied primarily in the Voice over IP technology. The latest version of ITU-T recommendation G.107 was released in December 2011 but the development of the E-model is not finished yet.

The second part of the paper deals with the proposed simplified E-model and its implementation in the Monitoring module of the BESIP project. In order to enable the comparison of results, PESQ was applied in accordance with ITU recommendation P.862, nowadays regarded as the standard for determining MOS. The fifth section of the paper addresses thee verification and results. With respect to the application of the speech quality assessment in practice, PESQ is not suitable for a routine operation as it is based on the evaluation the original and the degraded signals. On the other hand, the non-intrusive computational approach allows determining the speech quality without the need to compare both signals. Within the observed range of packet losses, the proposed model as described in Chapter 5.1 and its implementation as described in Chapter 5.3 achieves a decent correlation to PESQ.

I would like to stress two significant benefits of the presented applied research. The first is the fact, confirmed by the experiment, that the simplified E-model corresponds with the results obtained from PESQ for selected codecs which are included in BESIP, with minimal deviations and conformity higher than 98% for all codecs compared. The second contribution is the design and the practical implementation of the Monitoring module in BESIP which enables assessing the speech quality. The BESIP project is downloadable as open source and the published code can be a valuable inspiration for other developers.

Further improvements lie in the support of more codecs, acceleration and optimization of the implemented algorithm and especially in the notification of speech quality impairment. The notification of particular events is an inherent part of monitoring systems such as Zabbix or Nagios. Consequently, a log agent for the selected system could be developed in the near future.

Acknowledgement
The research leading to the results described in this paper has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 218086.

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] Methods for subjective determination of transmission quality, ITU-T Recommendation P.800, Geneva, 08/1996.
[3] Mean Opinion Score (MOS) terminology, ITU-T Recommendation P.800.1, Geneva, 07/2006.
[4] A. E. Mahdi, D. Picovici, Advances in voice quality measurement in modern telecommunications, *Digital Signal Processing*, Volume 19, Issue 1, January 2009, pp.79-103.
[5] M. Voznak, F. Rezac, M. Halas, Speech Quality Evaluation in IPsec Environment, In Proc. *12th International Conference on Networking, VLSI and Signal Processing (ICNVS '10)*, Cambridge UK, 2010, pp. 49-53.
[6] J. Kouril, H. Atassi, Objective Speech Quality Evaluation. A primarily Experiments on a Various Age and Gender Speakers Corpus, In Proc. *8th WSEAS International Conference on Circuits, Systems, Electronics, Control & Signal Processing*, Puerto de la Cruz, 2009 pp. 333-336.
[7] M. Voznak, Speech bandwidth requirements in IPsec and TLS environment, In Proc. *13th WSEAS International Conference on Computers*, Rhodes, 2009, pp.217-220.
[8] Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, ITU-T Recommendation P.862, Geneva, 02/2001.
[9] J. Rozhon, M. Voznak, Development of a speech quality monitoring tool based on ITU-T P.862, In Proc. *34th International Conference on Telecommunications and Signal Processing (TSP 2011)*, Budapest, 2011, pp. 62-66.
[10] 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, Computers,* Faro, November 2010, pp. 119-124
[11] 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.
[12] The E-model: A computational model for use in transmission planning, ITU-T Recommendation G.107, Geneva, 04/2009.
[13] 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.
[14] 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.
[15] Transmission impairments due to speech processing, ITU-T Recommendation G.113, Geneva, 11/2007.
[16] 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.
[17] M. Voznak, F. Rezac, Web-based IP Telephony Penetration System Evaluating Level of Protection from Attacks, *WSEAS TRANSACTIONS on COMMUNICATIONS* , Issue 2, Volume 10, February 2011, pp. 66-76.

[18] G. Cole, H. Rosenbluth, Voice over IP performance monitoring, *ACM SIGCOMM Computer Communication*, New York, 2001.
[19] Management of BESIP, last. rev. April 2012. URL: https://homeproj.cesnet.cz/projects/besip
[20] Source Code of BESIP, last. rev. April 2012. URL: http://liptel.vsb.cz/mirror/besip/nightly
[21] 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 (ICCCN 2011)*, 2011, DOI 10.1109/ICCCN.2011.6005933.
[22] M. Voznak, F. Rezac, VoIP SPAM and a defence against this type of threat, In Proc. *14th WSEAS International Conference on Communications*, Corfu, 2010, pp 172-177.
[23] F. Neri, Learning and Predicting Financial Time Series by Combining Evolutionary Computation and Agent Simulation, *Transactions on Computational Collective Intelligence*, Vol.6, Springer, Heidelberg, Vol.7, pp.202-221
[24] Н. К. Обжелян, В. Н. Трунин-Донской. *Речевое общение в системах "Человек-ЭВМ"* , Кишинев : Штиинца, 1985. - 175 с.