Real-Time Estimation of Speech Quality through the Internet Using Echo State Networks

Sebastián Basterrech and Gerardo Rubino

***Abstract*—Audio quality in the Internet can be strongly affected by network conditions. As a consequence, many techniques to evaluate it have been developed. In particular, the ITU-T adopted in 2001 a technique called Perceptual Evaluation of Speech Quality (PESQ) to automatically measuring speech quality. PESQ is a well-known and widely used procedure, providing in general an accurate evaluation of perceptual quality *by comparing the original and received voice sequences*.**

One obvious inherent limitation of PESQ is, thus, that it requires the original signal (we say the *reference*), to make its evaluation. This precludes the use of PESQ for assessing the perceived quality in real-time, as the reference is in general not available.

In this paper, we describe a procedure for estimating PESQ output working only with measures taken on the network state and properties of the communication system, without any use of the reference. It is based on the use of statistical learning techniques. Specifically, we rely on recent ideas for learning with specific types of neural networks, known under the name of Echo State Networks (ESNs), a member of the class of Reservoir Computing systems. These tools have been proven to be very efficient and robust in many learning tasks. The experimental results obtained show the good accuracy of the resulting procedure, and its capability to give its estimations of speech quality in a real-time context. This allows putting our measuring modules in future Internet applications or services based on voice transmission, for instance for control purposes.

***Index Terms*—Quality assessment, speech quality, echo state networks, reservoir computing.**

I. Introduction

Measuring the quality of a voice signal transmitted over the Internet is an important topic today, and one of main available tools for this purpose is the Perceptual Evaluation of Speech Quality (PESQ) method accepted in 2001 as the ITU-T objective speech quality measurement standard P.862 [1]. The network conditions vary over time, and in many contexts, several different factors lead to losses, which in turn lead to degradations in the perception of the quality by the users. PESQ analyzes this quality by comparing the received signal with the original speech sequence. For this reason, we say that it is a "full reference" technique, the reference being the original signal. Researchers in many areas use PESQ and the tool has been widely diffused in commercial measurement products. Recently, the ITU started to update its voice assessment recommendations by promoting the new P.863 standard Perceptual Objective Listening Quality Assessment (POLQA) [2], but PESQ still remains the main tool for voice quality assessment, and probably will continue to be so in the upcoming years. This is because it provides reasonably good correlation with the scores given by humans to VoIP applications. Observe that since PESQ requires the original signal, we cannot use it in real-time conditions.

In [3] the authors present a method for approximating the values given by PESQ but without any need for the original sequence. The idea is to estimate PESQ scores using a Feedforward Neural Network model, based on data concerning the packet loss process provoked by the network. These neural models are used because they are simple to manipulate and they lead to good results, even if another type of learning tool, the Recurrent Neural Network, exist. The latter are, in general, very powerful to learn non-linear mappings (which is the case in assessing perceptual quality) using sequential training algorithms. However, their use has been limited mainly due to the inefficiency of their learning algorithms [4], [5], which suffer from slow convergence rates and low robustness, thus in particular limiting their applicability in real-time contexts.

Recently, a new computational neural model has been proposed under the name of Reservoir Computing (RC). It offers a solution to the previously mentioned drawbacks of recurrent architectures while introducing no significant disadvantages. The first two proposed RC models were *Liquid State Machines* (LSMs) [6] and *Echo State Networks* (ESNs) [7], both almost simultaneously published. The two types of models have been successfully applied in many problems achieving very good results in temporal and non-temporal learning tasks [5], [8], [9].

In this paper, we study the problem of estimating PESQ scores in a context where the reference signal is not available, using the ESN model. The main idea is to capture the relation between certain network parameters that affect the perceived quality and their corresponding PESQ scores. The ESN tool is known for its modeling accuracy, parsimony and efficiency in the learning process [5]. Another notorious property is that the obtained tool is simple to extend or to update. Its extensibility and parsimony properties can be useful when new data is known when the system is already in operation. Our approach offers a new method for VoIP quality assessment in the context of Internet applications or services, which is able to provide accurate assessments in real-time. To illustrate the performance of our model, we present some numerical results, and we also add a couple of comparisons with other basic statistical learning techniques.

The paper is organized as follows. In Section II, we begin

Manuscript received March 1, 2013; revised April 15, 2013. This work was supported in part by the European Celtic Project "QuEEN".

S. Basterrech is with the University of Rennes 2, Rennes, France (e-mail: Sebastian.Basterrechtiscordio@etudiant.uhb.fr).

G. Rubino is with the National Institute for Research in Computer Science and Control (INRIA Rennes – Bretagne Atlantique), Rennes, France (e-mail: Gerardo.Rubino@inria.fr).

DOI: 10.7763/JACN.2013.V1.37

by describing the basic concepts, a network loss model and the PESQ algorithm. In Section III, we present RC together with the ESN method, our learning tool. Section IV presents the data collection details as well as the parameters used to ESN initialization and in the learning process. In Section V, we present and discuss the results obtained. Finally, we end with our conclusions and a discussion regarding future lines of research.

II. Some Concepts and Tools Related to Speech Quality Assessment

Speech Quality Assessment (SQA) methods are used to quantify speech quality of VoIP, for instance in commercial products. The ITU-T's PESQ algorithm [10] is the main tool to *objectively* evaluating speech quality over telecom networks (the other way is by performing *subjective tests* where panels of human users evaluate a set of sequences). In PESQ, the original and degraded signals are mapped onto an internal representation using a perceptual model. A kind of distance between the signals is computed using a cognitive model, to evaluate the perceived quality of the degraded signal [10]. The results are usually presented in the form of a Mean Opinion Score (MOS) (coming from its use in subjective testing), using a scale between 1 (Bad) and 5 (Excellent).

The approach of [3] to *replace* PESQ takes its roots from the idea of Pseudo-Subjective Quality Assessment (PSQA), a technique based on merging subjective quality assessment with methods of statistical learning [11], [12]. The idea is to observe that if quality degrades, this comes from the impact of many network impairments such as losses or delays, plus factors related to the application itself (codec used, bandwidth of the connection, etc.). Packets can be lost due to routing problems, network congestion, delays at intermediate points in the network (making packets arrive too late), etc. Among these, the loss process is in some sense the main affecting component with respect to the perceived quality [13], [14]: if everything arrives (and arrives in time), the receiver perceives exactly what is sent (assuming the speech signal perfect at the sender's side). In this paper, we focus on this loss process.

Following [3], we chose two metrics to characterize the loss process at the packet level: the *loss rate* (LR) and the *mean loss burst size* (MLBS). The first one refers to the ratio between the number of lost packets and the total number of transmitted packets, and the second one, as its name indicates, is the mean size of loss bursts (sequences of consecutive lost packets surrounded by two successfully transmitted ones) in the flow, thus a real number greater than or equal to 1. Both metrics are seen as *instantaneous*: we center a small window at time t and we measure both metrics inside the window. A comment here on delays: end-to-end delay is an important factor on the perceived quality in multimedia traffic, especially in interactive applications. This means that it is related to conversational quality, but does not influence voice quality itself. For this reason, we have chosen to ignore it in the present work. The variation of delay, called *jitter*, is another quality impacting measure. Unlike delay, this measure affects both interactive and one-way streams. When packets arrive too late at the receiver, they are discarded. As a consequence, these late packets can be counted as lost, and thus the effect of jitter is included in packet loss [15]. Last, some codecs do nothing when this happens, and others try to fill the gaps in the flows in some way. See next paragraph for details about how this was taken into account in this paper.

In our experiments we used the G.711 codec of ITU-T. The codec is a device in VoIP used to provide digitalization, compression and packetization. The codec includes a Packet Loss Concealment (PLC) algorithm to help hiding transmission losses in a packetized traffic. The PLC is used to mask packet losses in VoIP when they are not recoverable. In our experiments, we considered the case when this procedure was activated and the case where no technique was used when packets arrived late.

Several methods have been proposed to model burst losses in a range of communication channels. One of them is a simplified version of Gilbert's model [16]. Let us denote by X_n a binary random variable indicating if packet n is lost. We code $X_n = 1$ if the nth packet in the flow is lost, and $X_n = 0$ if the packet arrives at the receiver's player. The model consists in considering that $(X_n)_{n\geq 1}$ is a homogeneous Markov chain on the state space $\{0,1\}$. The two parameters of the model are the probabilities $p = \Pr(X_{n+1} = 1 | X_n = 0)$ and $q = \Pr(X_{n+1} = 0 | X_n = 1)$. We assume that $0 < p,\ q < 1$. This small chain is then ergodic. If (p_0, p_1) denotes its equilibrium vector, the balance equation at state 1, for instance, writes $p_1 q = p_0 p$, so

$$p_0 = \frac{q}{p+q}, \quad p_1 = \frac{p}{p+q}.$$

This means that the packet loss rate in equilibrium is $\mathrm{LR} = p_1 = p/(p+q)$. Observe that the size S of a burst of losses is a geometric random variable; its distribution is $\Pr(S = k) = (1-q)^{k-1} q,\ k \geq 1$, from which $\mathrm{MLBS} = 1/q$. Inverting these expressions we obtain

$$p = \frac{1}{\mathrm{MLBS}} \frac{\mathrm{LR}}{1-\mathrm{LR}}, \quad \text{and} \quad q = \frac{1}{\mathrm{MLBS}}.$$

One way of using this model is to measure LR and MLBS on traffic traces and then computing the model's parameters p and q using the given formulas.

III. Description of The Method Proposed

Our goal is to design a system tuned for the G.711 codec, where we measure LR and MLBS at the received side of a speech transmission, and we derive a MOS-like value of the perceived quality of the sequence. This value must be close enough (for networking purposes) to the value that PESQ would give to the received sequence after comparing it to the original one. Observe that since our measures are logically "instantaneous", if our estimation of PESQ scores is efficient enough, we will be able to estimate the perceived quality in real-time.

Let us start by discussing our learning module. Since the early 2000s a new approach for designing and training

recurrent neural networks has been investigated under the name of Reservoir Computing (RC). The *Liquid State Machines* (LSMs) and the *Echo State Networks* (ESNs) are considered the two principal references among RC models, respectively proposed by W. Maass [6] and H. Jaeger [7]. Both models have two well-differentiated structures. One is called *reservoir*, or *liquid* (depending if we use the ESN or the LSM model), and the other structure is called *readout*. The reservoir is a recurrent neural network where the neurons are interconnected by random and sparse weighted connections.

We have three types of non-zero weights in the model: between inputs and reservoir neurons, indicated by w_*^{in} below, among reservoir neurons, indicated by w_*^r, and those indicated by w_*^{out}, between input and readout units, and between reservoir and readout units.

The weights denoted w_*^{out} are updated using any type of classification or regression method. The idea of RC is to renounce training the reservoir, which means that they don't change the weights denoted w_*^r neither those denoted w_*^{in}. These two sets of weights are used as a nonlinear projection from the input data space into a new space having a higher dimension, with the objective of better "separating" data. Then, the learning side is concentrated on the readout part. This approach is based on the empirical observation that under certain hypothesis, a learning process restricted to the readout weights is often sufficient to obtain an excellent performance in many learning tasks [5]-[8].

The main difference between LSMs and ESNs lies in the type of nodes included in the reservoir. The LSM model comes from the interest in representing microstructures in the brain. In this context, the reservoir is built using spiking neurons [6], [17]. ESNs come from the field of Machine Learning. The reservoir in the ESN model is built using analog sigmoidal neurons. Even though spiking neurons have more computational power than sigmoidal neurons [18], we have chosen to use the ESN for estimating PESQ scores in real-time. In the past few years there has been a growing interest in the ESN model, since it has been proven efficient and robust in many machine learning benchmark problems [4], [5], [7]-[9], [19]-[21]. The device proposed in this paper is based on this tool, in order to exploit its simplicity and its robustness.

The activation function of the neurons in ESNs is generally $\tanh(\cdot)$. As stated above, the architecture of the network consists in three layers where the input and output layer are topologies without cycles. Recurrences only can be present in the reservoir. Denote by N_a the number of input units, by N_x the number of units in the reservoir and by N_b the number of outputs. In a learning process, we must approximate an unknown function $f(\cdot)$, here from $\Re^{N_a}$ into $\Re^{N_b}$. Our data is a set of K input-output pairs, called *training data*, and denoted $\left\{\left(\mathbf{a}^{(k)},\mathbf{b}^{(k)}\right):\mathbf{a}^{(k)}\in\Re^{N_a},\mathbf{b}^{(k)}\in\Re^{N_b},k=1,2,\mathbf{K},K\right\}$. This means that $f\left(\mathbf{a}^{(k)}\right)=\mathbf{b}^{(k)},k=1,2,\mathbf{K},K$. Let $\mathbf{w}^{in}$ be a $N_x\times(1+N_a)$ matrix which contains in its first row the bias terms and elsewhere the weights of the connections between input and reservoir units. The weight matrix $\mathbf{w}^r$ of dimensions $N_x\times N_x$ contains the weights of the connections among units in the reservoir. The adjustable weight matrix $\mathbf{w}^{out}$ of dimensions $N_b\times(1+N_a+N_x)$, contains the weights of the connections between input or reservoir units, and output ones. In the first row $\mathbf{w}^{out}$ contains the corresponding bias terms.

Each reservoir neuron m has a *state* denoted by $x_m\in\Re$. As a consequence, the output of the system depends on three components: the weights, which are parameters (fixed reals), the input $\mathbf{a}$ and the (internal) state $\mathbf{x}=\left(x_1,\mathbf{K},x_{N_x}\right)$ of the reservoir. When an input $\mathbf{a}$ is offered to the network, the network first updates its internal state $\mathbf{x}$, and then computes the output.

Let us make this explicit in our case. First, we must update the states. This is done in the following equation:

$$x_m^{new}=a\tilde{x}_m+(1+a)x_m \tag{1}$$

where

$$\tilde{x}_m=\tanh\left(w_{m0}^{in}+\sum_{i=1}^{N_a}w_{mi}^{in}a_i+\sum_{i=1}^{N_x}w_{mi}^{r}x_i\right). \tag{2}$$

Then, we make $x_m=x_m^{new}$ for each reservoir unit m. The network output $y_j,j=1,\mathbf{K},N_b,$ is calculated as follows:

$$y_j=w_{j0}^{out}+\sum_{i=1}^{N_a}w_{ji}^{out}a_i+\sum_{i=1}^{N_x}w_{ji}^{out}x_i, \tag{3}$$

where $a\in(0,1]$ is a parameter of memory control called *leaking rate*. The initial states of the neurons are basically arbitrarily chosen (for details, see [4], [5]).

It is better to see this system as a dynamical one when it is in operation. The network receives a sequence of inputs $\mathbf{a}(1),\mathbf{a}(2),\mathbf{K}$ and produces a sequence of outputs $\mathbf{y}(1),\mathbf{y}(2),\mathbf{K}$. For this, it computes a sequence of internal states $\mathbf{x}(1),\mathbf{x}(2),\mathbf{K}$ making that at time t, when computation starts the state of the reservoir is $\mathbf{x}(t-1)$ and the new state computed at that step t is $\mathbf{x}(t)$.

An observation here: our tool has as inputs LR and MLBS. In operation, these parameters are measured every Δ seconds, and the PESQ estimation is computed. In general, the output at time t (which is a logical time, corresponding to some real-time of the form $\tau_0+t\Delta$) is correlated with the output at the previous epoch, $t-1$. The ESN model has a structure allowing to better capture this effect, through the concept of state, as described above.

The RC approach consists in expanding the input pattern from $\Re^{N_a}$ into a larger space $\Re^{N_x}(N_a<<N_x)$ and in updating the weights in $\mathbf{w}^{out}$ during the training process. This matrix $\mathbf{w}^{out}$ is computed minimizing the Mean Square Error (MSE) between the target patterns $\left(b_1(t),\mathbf{K},b_{N_b}(t)\right)$ and the network outputs $\left(y_1(t),\mathbf{K},y_{N_b}(t)\right)$. The last ones are obtained using the input pattern $\left(a_1(t),\mathbf{K},a_{N_b}(t)\right)$. For simplicity, in this

work all weight matrices are initialized in a random way. It is probably possible to improve the experimental results using another initializing criterion [19], [20]. This is left for future work. We scaled the matrix $\mathbf{w}^r$ using its radius spectral to ensure good properties of the reservoir as it is recommended in [4], [5].

To better capture the performance of our technique, we compared its outputs (its predicted PESQ scores) with two commonly used regression models. Remember that our data are the two real numbers LR and MLBS, plus the binary variable PLC indicating if losses are considered by the player (PLC=1) for some correcting actions if possible, or if nothing is done (PLC=0). To handle this binary variable, we built two separated training data corresponding to the cases PLC=0 and PLC=1. The first one, model A, is a simple linear regression with $N_a + 1$ unknown parameters. The two input variables a_1 and a_2 correspond to the parameters LR and MLBS, respectively. Predicted values are computed as follows:

$$y_A(t) = b_0 + b_1 a_1(t) + b_2 a_2(t).$$

The second model B uses a linear combination proposed in [22] to estimate VoIP quality. The authors used other independent variables to make the prediction (the delay and the packet loss rate). Predicted values are computed using the following expression:

$$y_B(t) = g_0 + g_1 a_1(t) + g_2 a_2(t) + g_3 a_1^2(t) + g_4 a_2^2(t) + g_5 a_1(t) a_2(t).$$

The regression parameters b_i with $i = 0,1,2$ and g_j for $j = 0,1,2,3,4,5$ were calculated by means of the R tool.

IV. Experimental Setup

The experimental setup used for this study was inspired by the setup method in [3]. We used G.711 encoding, with and without loss concealment. Our network parameters are LR and MLBS. The network loss model used is the simplified Gilbert model [16] described in Section II. The range of LR and MLBS values was chosen considering very extreme scenarios and covering all possible loss conditions in VoIP problems. PESQ can produce poor results when the degraded signals are subject to large loss instances. In network conditions where the impairments are too high, poor correlations between PESQ and subjective scores can arise. Since the goal of this work is to built a mapping between PESQ scores and learning methods, we considered a very large loss space anyway. We considered a range of loss rate values containing integers between 1% and 30% and mean loss burst sizes ranging from 1 to 6 packets. Table I shows the possible values of each dependent variable of the predictive model.

TABLE I: Empirical Setup for Packet Loss Behavior Using the Simplified Gilbert Model [16]

Variable	Values
LR	1% , 2%, 3%, …, 30%
MLBS	1, 1.25, 1.5, 1.75, 2, 2.5, 3, 3.5, 4, 5, 6
PLC	0, 1

Given that standard-length (approximately 10s) samples were used, it was not possible to have all combinations of LR and MLBS, since some of them are not possible within the approximately 400 packets that each speech sample contains. Thus, we considered only valid combinations. Following this, each loss trace created was verified to ensure that it had the desired characteristics.

We used the same data as in [3], starting with 20 standard speech samples, spoken in English, 50% by males and 50% by females. For each combination of values of the two loss-related parameters LR and MLBS, and for each original sequence, we generated 10 different traces (all statistically similar). In order to mitigate any dispersion when PESQ's results showed more variability, we used more samples in those cases. This means that for each vector (LR, MLBS, PLC) several sequences were analyzed (around 200 of them, except in some cases with high loss rates, where more samples were generated and used). In other words, in order to obtain PESQ scores we sent each original sequence through a simulated/emulated network varying the three considered variables LR, MLBS, PLC. Then we used PESQ to evaluate the resulting quality. For each point (LR, MLBS, PLC), many different associated PESQ scores were thus obtained (around 200 of them). We constructed a data set containing roughly 128500 entities of network parameters and their PESQ evaluations.

We then built a second smaller table having around 600 entries where each row corresponds to a point (LR, MLBS, and PLC) and the median of the PESQ values corresponding to this point in the larger table. In this, we followed the proposal in [15]. Following the standard procedures in statistical learning, we randomly (and uniformly) separated the data set into two subsets: a training and validation set. The first one contains 80% of the data was used for training the model. The rest of the data was used to validate the training process' result. Figure 1 illustrates PESQ evaluation approach using network impairments.

To evaluate the accuracy of the predictor, we used the *Mean Square Error* (MSE), which measures how far is the prediction from the target data. If the validation set has K' pairs, assumed indexed from 1 to K', the MSE is:

$$E = \frac{1}{K'} \sum_{k=1}^{K'} \left(y^{(k)} - b^{(k)} \right)^2,$$

where $y^{(k)}$ is the model's prediction when the input is $\mathbf{a}^{(k)}$.

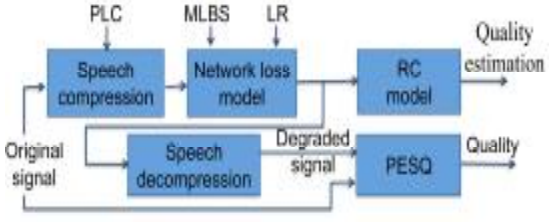

Fig. 1. Schema of how to build a RC model to predict PESQ scores using network impairment information.

We used an ESN with a large, sparsely and randomly connected reservoir as suggested in the literature [5]. We experimented with different reservoir sizes, containing between 10 and 50 units. In many ESN applications it has

been shown that the reservoir size is an important parameter [19]-[21]. As in all learning systems, there is a tradeoff to reach in that parameter. If it is too small, we don't exploit enough the benefits of separating the set of possible inputs inside a larger space, and we can have poor training processes. If it is too large, training is easy but generalizing can be very bad (the usual overfitting phenomenon). We present the experimental results for a reservoir size with 30 units. This size was chosen considering that the best model occurs when the validation error has its global minimum. The weight matrices $\mathbf{w}^{in}$, $\mathbf{w}^{r}$ and $\mathbf{w}^{out}$ were randomly initialized from a uniform distribution over an interval [-0.5, 0.5]. After that, for each weight a binary random trial was done, and with probability 0.8 the connection was removed (that is, the weight was set to 0). Following a rule of thumb discussed in [5], we scaled $\mathbf{w}^{r}$ to obtain a matrix with spectral radius equal to 0.1. Last, we used a leaky rate = 0.9.

V. Results

We begin by scaling the data set into the range [0, 1]. The results are also presented in the [0, 1] range. Table II provides the performance of the three methods employed. Given the fact that we are using scaled data, the reached MSE using the ESN model is extremely small. The dispersion of the data when PLC=1 is smaller than PLC=0. This affects the accuracy of the learning procedure. For this reason, when PLC=1 the MSE is smaller than in the other case. Figure 2 and Figure 3 show the accuracy of ESN with 2 input neurons, 30 neurons in the reservoir and 1 output neuron for the data set when PLC=0 and PLC=1 respectively. In the *y*-axis we put the PESQ estimation using ESN and in the *x*-axis we put the median of the PESQ scores obtained for each point (LR, MLBS, PLC). Fig. 4 shows PESQ values and estimations versus LR with MLBS=0.5 and PLC=0. In the graph there are two types of spots: black and red. Each black spot in the graphs corresponds to (LR, 0.5, 0). Each red point corresponds to the PESQ estimation made by our tool for points (LR, 0.5, 0). Observe that for each vector (LR, 0.5, 0), we have a predicted PESQ value (the red spot) and around 200 measured PESQ values (the black spots). To obtain more visibility we draw a curve through the sequence of red points, each pair of consecutive red spot being connected by a line. In the same way we built Figure 5, where the points used are (LR, 0.2, 1) and their PESQ scores. Both figures show the importance of LR in the PESQ algorithm. In both situations our PESQ estimation is very close to the median of the PESQ scores. All figures illustrate that PESQ scores are higher when PLC=1 and that the accuracy of the predictive model is better in this case.

TABLE II: Accuracy Comparison among the Three Models Considered

Method	MSE (PLC=0)	MSE (PLC=1)
Model A	0.002702	0.001561
Model B	0.000770	0.000389
ESN	0.000334	0.000214

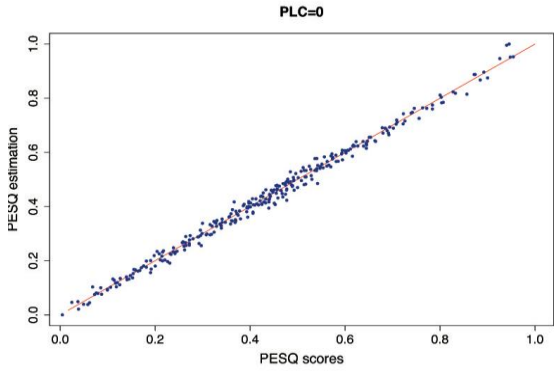

Fig. 2. Performance of PESQ scores estimation using ESN with 30 units in the reservoir. The data is scaled in [0,1]. Plot shows average of PESQ values versus the PESQ estimation for each pair (LR, MLBS) when data is obtained with packet loss concealment inactive (PLC=0).

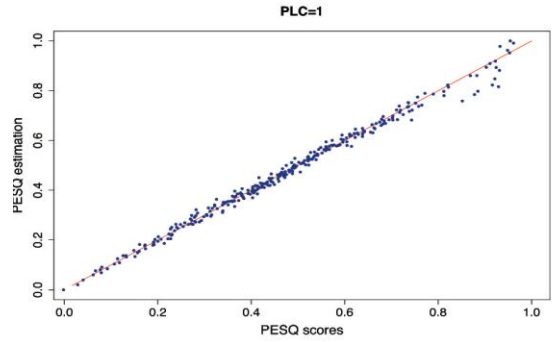

Fig. 3. Performance of PESQ scores estimation using ESN with 30 units in the reservoir. The data is scaled in [0,1]. Plot shows average of PESQ values versus the PESQ estimation for each data pair (LR, MLBS) when data is obtained with packet loss concealment (PLC=1).

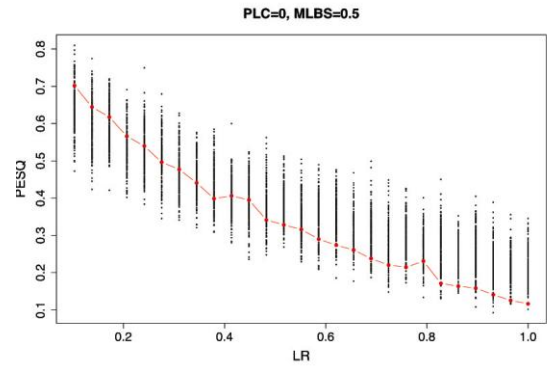

Fig. 4. The PESQ scores are measured for input data with PLC=0 and MLBS=0.5. Each black spot corresponds to a specific measure of PESQ and the red curve shows the PESQ estimation using ESN.

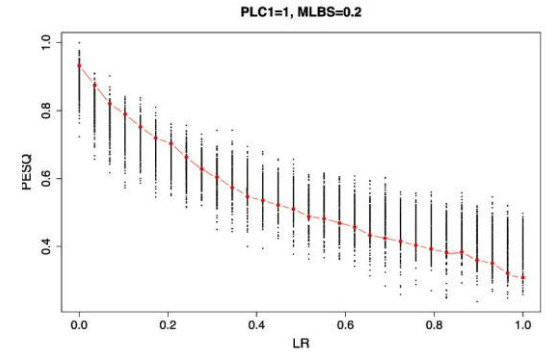

Fig. 5. PESQ scores measured for input data with PLC=1 and MLBS=0.2. Each black spot corresponds to a specific measure of PESQ and the red curve shows the PESQ estimation using ESN.

VI. Conclusions

In this paper, we propose a procedure that can replace the PESQ tool for evaluating the quality of a speech signal as perceived by the receiver of an Internet connection. The interest of the procedure is that it does not need to access the original sequence, as sent by the sender, which is the case for PESQ. The device proposed is based on the use of statistical learning techniques. More specifically, we used Echo State Networks, which are among the most widespread and successful Reservoir Computing models. Moreover, our method can work in real-time. This makes it appropriate for networking applications, for instance, for network control.

In the close future we will explore the capability of our technique to continue learning while it is in operation, exploiting in this way one of the main properties of the learning method used, the Echo State Network.

References

[1] J. G. Beerends, A. P. Hekstra, A. W. Rix, and M. P. Hollier, "Perceptual Evaluation of Speech Quality (PESQ) the new ITU Standard for end-to-end Speech Quality Assessment Part II: Psychoacoustic model," *Journal Audio Engineering Society*, vol. 50, no. 10, pp. 765-778, 2002.

[2] ITU-T Recommendation P.863. Perceptual Objective Listening Quality Assessment. (2011). [Online] Available: http://www.itu.int/rec/T- REC- P/en

[3] S. Basterrech, G. Rubino, and M. Varela, "Single-sided real-time PESQ score estimation," *International Proceeding of Measurement of Speech, Audio and Video Quality in Networks (MESAQIN'09)*, Prague, Czech Republic, June 2009, pp. 94-99.

[4] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and applications of Echo State Networks with leaky-integrator neurons," *Neural Networks*, vol. 20, no. 3, pp. 335-352, April, 2007.

[5] M. Lukoševičius and H. Jaeger, "Reservoir Computing approaches to Recurrent Neural Network training," *Computer Science Review*, vol. 3, pp. 127-149, 2009.

[6] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: a new framework for a neural computation based on perturbations," *Neural Computation*, vol. 14, no. 11, pp. 2531-2560, 2002.

[7] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks," German National Research Center for Information Technology, Technical Report, no. 148, 2001.

[8] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural Networks*, vol. 20, no. 3, pp. 287-289, April, 2007.

[9] M. Lukoševičius, H. Jaeger, and B. Schrauwen. (2012). Reservoir Computing Trends. *KI-Künstliche Intelligenz*. [Online]. 1-7. Available: http://dx.doi.org/10.1007/s13218-012-0204-5

[10] ITU-T Recommendation P.862. Perceptual Evaluation of Speech Quality (PESQ), an Objective Method for End-To-End Speech Quality Assessment of Narrowband Telephone Networks and Speech Codecs. (2001). [Online]. Available: http://www.itu.int/rec/T-REC- P/en

[11] G. Rubino, "Quantifying the Quality of Audio and Video Transmissions over the Internet: the PSQA Approach," in J. Barria, editor, *Design and Operations of Communication Networks: A Review of Wired and Wireless Modelling and Management Challenges*. Imperial College Press, 2005.

[12] S. Mohamed, G. Rubino, and M. Varela, "Performance evaluation of real-time speech through a packet network: a Random Neural Networks-based approach," *Performance Evaluation*, vol. 57, no. 2, pp. 141-161, 2004.

[13] 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, no. 6, pp. 1957-1968, November, 2006.

[14] A. W. Rix, J. G. Beerends, D.-S. Kim, P. Kroon, and O. Ghitza, "Objective assessment of speech and audio quality technology and applications," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 14, no. 6, pp. 1890-1901, 2006.

[15] M. Varela, I. Marsh, and B. Grönvall, "A systematic study of PESQ's performance (from a networking perspective)," presented at Measurement of Speech and Audio Quality in Networks workshop (MESAQIN'06), Prague, Czech Republic, June, 2006.

[16] E. Gilbert, "Capacity of a burst-loss channel," *Bell Systems Technical Journal*, vol. 5, no. 39, September,1960.

[17] W. Maass, T. Natschläger, and H. Markram, "Computational models for generic cortical microcircuits," *Neuroscience Databases, A Practical Guide*. Boston, Usa: Kluwer Academic Publishers, June 2003, pp. 121-136.

[18] W. Maass. (1999). Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons. Institute for Theoretical Computer Science. Technische Universitaet Graz. Graz, Austria, Technical Report TR-1999-037. [Online]. Available: http://www.igi.tugraz.at/psfiles/90.pdf

[19] M. Lukoševičius, "On self-organizing reservoirs and their hierarchies," Jacobs University, Bremen, Technical Report no. 25, 2010.

[20] S. Basterrech, C. Fyfe, and G. Rubino, "Self-Organizing Maps and Scale-Invariant Maps in Echo State Networks," in *Proc. IEEE Conf. Intelligent Systems Design and Applications (ISDA)*, Spain, November, 2011, pp. 94-99.

[21] A. Rodan and P. Tino, "Minimum Complexity Echo State Network," *IEEE Transactions on Neural Networks*, vol. 22, no. 1, pp. 131-144, 2011.

[22] L. Sun and E. Ifeachor, "Voice quality prediction models and their application in VoIP networks," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 809-820, August, 2006.

Sebastián Basterrech received a M.Sc. degree in Applied Mathematics in 2008 from the Aix-Marseille University, France. From 2009 to 2012 he received a doctoral fellowship from the National Institute for Research in Computer Science and Control (INRIA), France. He obtained the Ph.D. degree in Computer Sciences in 2012 from the University of Rennes 1, France. From 2011 to 2013 he was a M.Sc. student in Computer Arts at the University of Rennes 2 in France. Since June 2013 he is a postdoctoral researcher at the VŠB-Technical University of Ostrava, Czech Republic. The main research interests of S. Basterrech include spatio-temporal data mining, reservoir computing and bio-inspired algorithms for engineering applications.

Gerardo Rubino is a senior researcher at INRIA (the French National Institute for Research in Computer Science and Control) where he is the leader of the DIONYSOS (Dependability, Interoperability and perfOrmaNce analYsiS of netwOrkS) team. His research interests are in the quantitative analysis of computer and communication systems, mainly using probabilistic models. He is the author of the PSQA technology for automatic assessment of perceived quality of voice or video. He also works on the quantitative evaluation of perceptual quality of multimedia communications over the Internet. He is a member of the IFIP WG 7.3.