ANALYSIS OF DIFFERENT ACOUSTIC FRONT-ENDS FOR AUTOMATIC VOICE OVER IP RECOGNITION

D. Falavigna[(1)], M. Matassoni[(1)], S. Turchetti[(2)]

(1) - ITC-irst, Via Sommarive 18, 38050 Povo, Trento - Italy
(2) - Department of Information Engineering, Via Gradenigo 6/B, 35131, Padova - Italy

ABSTRACT

We investigated the usage for automatic speech recognition of different acoustic features, obtained from the output bitstream of a voice over IP codec. In particular, we analyzed the influence, on recognition performance, of both analysis rate and vector quantization of acoustic parameters introduced by the codec. Particular care has to be taken to train acoustic models at the reduced analysis rate employed by the codec: some related issues are discussed in the paper. We also used a model for simulating packet loss and we measured the corresponding performance degradation. Experiments were carried out on both clean and noisy speech databases.

1. INTRODUCTION

The work reported in this paper has been motivated by the need of reducing the overall bit rate over the communication channels used in the multimodal browsing architecture presented in [1]. This last one consists of a web server, having Automatic Speech Recognition (ASR) and Text To Speech (TTS) capabilities, that can handle requests coming from several types of remote clients. Presently, the client devices send the speech samples to the server without using any coding; however, due to the limited bandwidth of the transmission channels, it is often necessary, in order to avoid long delays during the interaction, to greatly reduce the bit-rate in the client-server communication. The simplest way to do this is to use some existing voice codecs, possibly designed for transmitting voice over networks employing the Internet transmission Protocol (IP).

Note that for speech recognition purposes is not necessary to fully reconstruct the speech signal at the receiver, since only the acoustic parameters, in principle the spectral features and the energy, used by the recognizer can be transmitted. This last approach, called Distributed Speech Recognition (DSR), is proposed in [2] [3] and is one the most relevant research issues inside the speech recognition community.

This work has been partially funded by the EU Project Homey, IST-2001-32434

The objective of the present work is not to propose a novel approach for DSR, not even some related algorithms, but to analyze the features transmitted by a well known Voice over IP (VoIP) codec (G723.1) and measure the influence of the corresponding coded parameters on speech recognition performance. We focused on G723.1 [4] since it is one of the most commonly used codecs for VoIP transmission, due to the high compression rate it provides (5.3 or 6.3 kbit/sec) and to the quality of the decoded speech.

G723.1 is a CELP type (Code Excited Liner Predicted) codec. It transmits two types of information, to be used at the receiver for synthesizing the speech signal: Linear Predictive (LP) parameters, that account for the frequency response of the synthesis filter, and the excitation signal to the synthesis filter itself. The analysis at the coder side is carried out at a rate $7.5\ ms$, while the real transmission rate of the parameters is $30\ ms$ (see section 2 for the details). The frame energy is not explicitly transmitted, but is encoded through a combination of gains related to both the periodic component (pitch predictor gains) and non periodic component (pulse gains) of the excitation signal estimated at the encoder side.

In [5] is suggested to use, as observation vectors for the recognizer, acoustic parameters "directly" derived from the LP features transmitted by the codec (this allows to avoid the distortion introduced by the decoding process). Nevertheless, the approach has proven to be effective only in the presence of packet loss (see Table 4 of above mentioned reference [5]). This fact can be explained considering that the codec performs an analysis by synthesis on the residual signal, so that useful information is still contained in the transmitted excitation. Actually, works reported in [6] and [7] propose to use both coded LP features and features derived from the residual signal. This approach results in a significant performance improvement, showing that bitstream-based features are really effective only if also the residual signal is used to evaluate them. However, we found that this result depends on the analysis rate employed at the receiver, i.e. it is no longer valid at the analysis rate ($30\ ms$) really transmitted by the codec (the experimental details are given in section 4).

Similarly to what proposed in the above mentioned works, we trained different sets of Hidden Markov Models (HMM), using features derived from both the decoded speech signal or from the output bitstream of the codec. Furthermore, to cope with the larger analysis rate used by the codec ($30\ ms$ compared with $7.5\ ms$ of our baseline system), some changes in both the ASR front-end and HMM topology have been introduced with respect to the baseline system (see section 2 for the details).

A set of experiments has been carried out on two different databases: *APASCI* and *AURORA2*. *APASCI* [8] is an Italian phonetically rich database recorded in a clean environment, *AURORA2* [2] is a noisy American-English database formed by digit utterances. For each speech waveform in the two databases three signals have been considered in the experiments: the original one, the corresponding encoded stream and the decoded signal.

As one can expect, encoded speech gives lower recognition performance compared to the original one (we measured, on the $AURORA2$ task, an increase of about 30% in the relative word error rate). This performance drop is mainly due to the reduced analysis rate employed by the codec while, on the contrary, vector quantization of the acoustic features has much lower influence on the overall word error rate. Furthermore, decoded speech exhibits better performance than the encoded one at $7.5\ ms$ analysis rate, meaning that, as seen above, useful information is contained in the residual signal. Finally, we noted that improvements can be obtained if the encoded LP parameters are linearly interpolated before being sent to the recognizer.

A further problem when dealing with packed switched networks is represented by "packet loss". Packets can be lost either because of network congestion or because of large transmission delays. We led a set of experiments simulating packet loss with a two state HMM (see section 3 for the details): output probability density functions and transition probabilities of the HMM allow to control both the packet loss rate and mean duration of the lost packets. Results are reported as a function of both packet loss rate and mean burst duration.

A final observation concerns the possibility to use discrete HMMs, directly trained on the codewords transmitted by the codec. Nevertheless, in this case it is not clear how to introduce in the acoustic observations an information similar to the time derivatives of the continuous case.

2. ACOUSTIC MODELING

The adopted codec operates on a telephone bandwidth signal, sampled at $8\ kHz$ and converted to 16-bit linear PCM. The encoder processes the speech signal by buffering consecutive frames of 240 ($30\ ms$) samples. Each frame is divided into 4 sub-frames having a length of 60 samples ($7.5\ ms$) each. Then, a 10-th order LP analysis is applied to Hamming windows of 180 samples ($22.5\ ms$), centered on each sub-frame, thus resulting into four 10-dimensional vectors of LPC coefficients for each analyzed frame. Only the LPC coefficients of the last (fourth) sub-frame are converted to Line Spectral Pairs (LSP), vector quantized and transmitted to the decoder. The excitation signal is obtained by means of an analysis by synthesis method and contains both a periodic component (it involves the estimation of the pitch period at the encoder) and an aperiodic one. Unlike LSP coefficients, the codes related to the excitation signal are transmitted for all the four sub-frames ([4]), resulting in an excitation frame rate of $7.5\ ms$.

To measure the loss of performance due to the coding process we led a set of experiments on the following three speech databases:

a) the given **original** ones, i.e. $APASCI$ or $AURORA2$;

b) the corresponding **encoded** databases, obtained by processing the original signals of $APASCI$ or $AURORA2$ with the G723.1 coder;

c) the related **decoded** databases, obtained by processing the original signals of $APASCI$ or $AURORA2$ with the G723.1 coder, followed by the corresponding decoder.

Comparisons have been made at two different analysis rates, i.e. $30\ ms$ and $7.5\ ms$, using observation vectors formed by 10 LPC cepstral coefficients, log-energy and corresponding first and second order time derivatives.

LPC Cepstral coefficients (LPCCs) of both original and decoded signals have been evaluated on Hamming windows of length $22.5\ ms$ (180 samples) by means of a 10-th order autocorrelation analysis. Instead, LPCs used in the experiments with encoded streams have been directly obtained from the transmitted LSPs. In this last case, since the encoded LSP parameters have been derived at a rate of $30\ ms$ (three sub-frames, out of four, are discarded at the encoder), a linear interpolation has been applied to them in order to simulate an analysis rate of $7.5\ ms$ for the encoded data. The adopted linear interpolation method is the one used by the codec, which only involves two frames: the current and the previous ones.

Since the frame energy is not explicitly encoded in the output bitstream of the codec we decided to derive it from the excitation signal reconstructed at the decoder. In fact, the gain information is coded together with information on both pulse positions and signs of the excitation. We have observed that speech recognition performance does not change whether the energy is evaluated on the excitation or on the decoded signal.

The HMMs used in the experiments correspond to a set of context independent phone units. In addition, we have introduced two specific units for modeling either the background noise and extra-noises, such as: telephone bursts, clicks, and so on. HMMs consists of 3 states left-to-right Markov chains with output distributions defined by a mixture of 16 Gaussian functions having diagonal covariance matrices. Furthermore, with a frame rate of 30 ms, skips among states are inserted in order to be able to train HMMs correctly. In fact, many phone units in the training database are shorter than the minimum length allowed by a 3 state model (i.e. $30 \times 3 = 90\ ms$).

For each one of the above reported types of speech data (original, encoded and decoded) a corresponding set of phone HMMs has been trained, while test has been carried out in both "matched" and "non matched" conditions.

3. SIMULATION OF PACKET LOSS

The G723.1 decoder employs an "error concealment" procedure for generating signal frames corresponding to packets missed during transmission. Basically, when the decoder realizes that packets have been lost, it sets both the actual filter parameters and excitation signal to the ones of the last correct frame. This process continues attenuating the output signal for at most 3 packets, after which the lost frames are replaced by silence (see [4] for more details).

The effects of packet loss on ASR performance have been evaluated using the model proposed in [9] to simulate the behavior of transmission channels with memory. The model, also known as Gilbert-Elliott channel model, is shown in Figure 1.

The model is a two state HMM: state G ("good"state) corresponds to a low packet loss condition, state B ("bad" state) corresponds to a high packet loss. The model is used in a generative way: the values of two uniformly distributed random variables determine the probabilities, P_G and P_B, to loose packets in the good and bad state, respectively (we assume $P_B >> P_G$). Similarly, transitions between state G and B and vice versa are governed by probabilities $P_{GB} = P[B|G]$ and $P_{BG} = P[G|B]$, respectively. This way, the probability to remain in state G is $P_{GG} = P[G|G] = 1 - P_{GB}$, while $P_{BB} = P[B|B] = 1 - PBG$ is the probability to remain in the bad state. Setting $P_{GB} << 1 - P_{BG}$ we can simulate the generation of burst errors. In fact, with a low value of P_{GB}, it is not likely to move from the good state to the bad one, but once the bad state is reached it is likely to stay there for many time intervals. By appropriately choosing the values of P_G, P_B, P_{GB}, P_{BG}, it is possible to simulate different conditions of packet loss.

Since the state duration probability density is an exponential function [9], the mean state permanence times, D_G and D_B, in states G and B respectively, are given by the following equations:

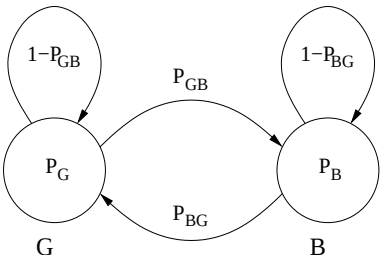

Fig. 1. *Model used for simulating packet loss.*

$$D_G = \frac{1}{1 - P_{GG}} \ , \ D_B = \frac{1}{1 - P_{BB}} \qquad (1)$$

Given the assumptions above, i.e. $P_B >> P_G$ and $P_{GB} << P_{BB}$, it results that the the average burst length is approximately given by D_B, the average time in which the system is in state B.

The packet loss probability P_L can be estimated with the following equation:

$$P_L = \frac{D_G}{D_G + D_B} \cdot P_G + \frac{D_B}{D_G + D_B} \cdot P_B \qquad (2)$$

We evaluated speech recognition performance against different values of both packet loss rates and mean burst lengths, using the model of Figure 1 to generate the desired types of packet loss in the chosen test databases.

4. EXPERIMENTS AND RESULTS

All the reported experiments have been carried out on both the $APASCI$ and $AURORA2$ databases.

$APASCI$ [8] is an Italian corpus, collected in our labs with the purpose of training and testing acoustic models for speech recognition. It consists of 2170 phonetically rich sentences, uttered by 100 speakers. We used 1950 sentences for training, 198 sentences for test and the remaining 22 sentences as development set. The database has been collected in a quiet room at 16 kHz sampling frequency. Therefore, all signals have been down-sampled to 8 kHz before being analyzed as explained in section 2.

For the $APASCI$ task, recognition grammar consists of a word loop graph: the number of word transitions in the graph is 2195, furthermore, specific transitions for modeling noises of various types have been introduced. Note that the chosen task is purely acoustic: that is no language model is used. However, to balance insertion and deletion errors a penalty is added to the total accumulated log-likelihoods, evaluated during the Viterbi search, before entering a word transition. The value of the penalty is chosen as the one that maximizes the Word Accuracy (WA) on the development set.

$AURORA2$ [2] is an American English corpus of both continuous and isolated digits: specifically, a filtered and down-sampled (at $8\ kHz$) version of part of the TIDigits database. The training part is formed by 8440 sentences uttered by 110 speakers, while the test part consists of 4004 sentences uttered by 104 speakers. Four different types of noises, at 5 different SNRs, have been added to the training set, resulting into a *multi-condition* training mode. Three test sets, namely A, B and C, have been defined. Test A contains utterances where the added noises (at different SNR) are of the same types of the ones of the training set, whereas noises used for Test B are of different types. Test C contains utterances in which both speech and noises have been filtered before being added. Refer to [2] for more details about the definition of both multi-condition training and test sets of $AURORA2$.

For $AURORA2$ the number of words to recognize is 11 (i.e. the ten digits plus "ow"). Since a development set is not available, we decided to balance insertion and deletion errors by directly acting on the topology of the recognition loop graph. More specifically, each digit in the graph is represented by multiple transitions having associated different penalties: the best transition penalties are automatically selected during the decoding phase. To train HMMs for $AURORA2$ we employed both clean and noisy data (i.e. the multi-condition training defined for $AURORA2$), while test has been conducted on the reference test sets A, B and C and performance has been averaged among them. As seen above, and contrary to what proposed in [2], the HMMs used in the experiments with $AURORA2$ correspond to phone units, not to digit words, while the adopted recognizer is the one developed in our labs [8].

Table 1 shows WAs obtained on the $AURORA2$ task with analysis rates of both 7.5 and $30\ ms$. The labels **orig**, **enc** and **dec** refer to the original, encoded and decoded signals respectively.

Table 1. *Word Accuracies obtained on the AURORA2 task (multi-condition training).*

	7.5 ms	30 ms
orig	**88.6%**	85.6%
dec	87.1%	84.9%
enc	85.8%	85.0%

The baseline performance corresponds to the WA obtained on the original data at $7.5\ ms$ frame rate. Note the large performance decrease, about 30% reduction of relative WA, between the baseline (88.6%) and the decoded voice (84.9%). This can be explained considering that the decoded speech has been subjected to distortions due to both coding, involving decimation and quantization of spectral parameters, and decoding processes.

Comparing the performance loss due to the larger analysis step ($30\ ms$ vs. $7.5\ ms$), we note that the effects of decimation are less for encoded data (from 85.8% to 85%, about 5% reduction in the relative WA) with respect to both original (from 88.6% to 85.6%, 26% relative WA reduction) and decoded (from 87.1% to 84.9%, 17% relative WA reduction) speech. This can be explained by considering that features encoded at $7.5\ ms$ are not really evaluated from the signal, but are obtained by linear interpolation of the transmitted (at a rate of $30\ ms$) LSP coefficients (see section 2). Nevertheless, third raw of Table 1 shows that linear interpolation of encoded parameters gives significant benefits; furthermore, as reported in [5], an additional improvement could be obtained by employing interpolation windows larger than two frames.

At the analysis rate of $7.5\ ms$, the decoded database exhibits better performance than the encoded one: 87.1% with respect to 85.8%. This suggests, as mentioned above, that the transmitted excitation signal, used to generate the output signal at the receiver, contains additional useful information for speech recognition. On the contrary, this trend is not observed at $30\ ms$ analysis rate, where "decoded" and "encoded" parameters exhibits similar performance (compare in Table 1 85.0% WA vs. 84.9% WA). Actually, parameter decimation at $30\ ms$ disregards three, out of four, sub-frames conveying residual signal information, greatly reducing the benefits it causes. In any case, above results suggest that the usage of the residual signal for deriving acoustic parameters is not effective at low frame rates, e.g. the one ($30\ ms$) used by the codec.

Comparing the baseline performance with the one obtained on the original database at $30\ ms$ frame rate (i.e. 88.6% vs. 85.6%), we observe a relative WA decrease of about 26%. This drop is quite completely attributable to the reduction of the analysis rate. On the other hand, WAs measured by passing from original to encoded data, at $30\ ms$ frame rate (compare 85.6% vs. 85.0%), exhibits a much smaller decrease indicating that quantization of acoustic parameters has not dramatic effects on WA.

Finally, we carried out some experiments in "non matched" training-test conditions (e.g. original vs. encoded data, encoded vs. decoded, etc.) obtaining, as one can expect, large performance losses.

On the $APASCI$ task we led a set of experiments similar to $AURORA2$. Results are given in Table 2.

Performance exhibits a similar trend to that of Table 1, but differences among the various types of signals are more marked. This is probably due to either a sort of "masking" effect, introduced by noise in $AURORA2$, and to the much smaller perplexity of $AURORA2$ (11 loop words) with respect to $APASCI$ (about 2000 loop words). However, note that results of Table 1 should be considered more consistent, from a statistical point of view, than those of Table

Table 2. *Word Accuracies obtained on the APASCI task.*

	7.5 ms	30 ms
orig	**60.4%**	50.9%
dec	53.5%	45.2%
enc	50.7%	47.7%

2, due to the much larger size of the $AURORA2$ test set with respect to $APASCI$.

4.1. Influence of packet loss

As explained in section 3, generation of bursts in the transmitted stream is simulated with the model of Figure 1 (as seen, bursts are produced in state B, i.e. the bad state of the model).

From equation 1, we observe that the Mean Burst Length ($MBL = D_B$), is only a function of the transition probability, $P_{BG} = 1 - P_{BB}$. On the contrary, Packet Loss Rate (PLR) depends upon all the probabilities involved in the model.

Experiments have been carried by selecting, according to equation 1, three different values of MBL, namely: 1.2, 2.2 and 4.0. The error probabilities, P_G and P_B, have been fixed at values 0.01 and 0.95, respectively. Finally, for each MBL, PLR is varied by changing, according to equation 2, the value of the transition probability P_{GB}.

Above values for MBL have been chosen in accordance with the error concealment procedure of the codec; as seen in section 3 the codec interpolates over an interval spanning from 1 to at most 3 frames. These values are also used in [5], and are suggested by previous studies and measurements [10] on packet loss over the Internet.

Figure 2 shows word accuracies obtained on both encoded (Enc) and decoded (Dec) $APASCI$ databases. In the Figure, word accuracies are given as functions of PLR, at different MBL values.

As one can expect, the overall performance decreases as PLR increases. Comparing the curves of figure 2, we also observe that WAs are less sensitive to the increase of MBL with respect to the increase of PLR. Furthermore, the relative WA decrease is higher for signals analyzed at 7.5 ms with respect to signals analyzed at 30 ms. This is probably due the fact that at 30 ms the system has "little to lose", while at 7.5 ms the contribution brought by the excitation signal quickly vanishes as frames are lost. Finally, note that performance loss is generally not high with small values of PLR (below 1%).

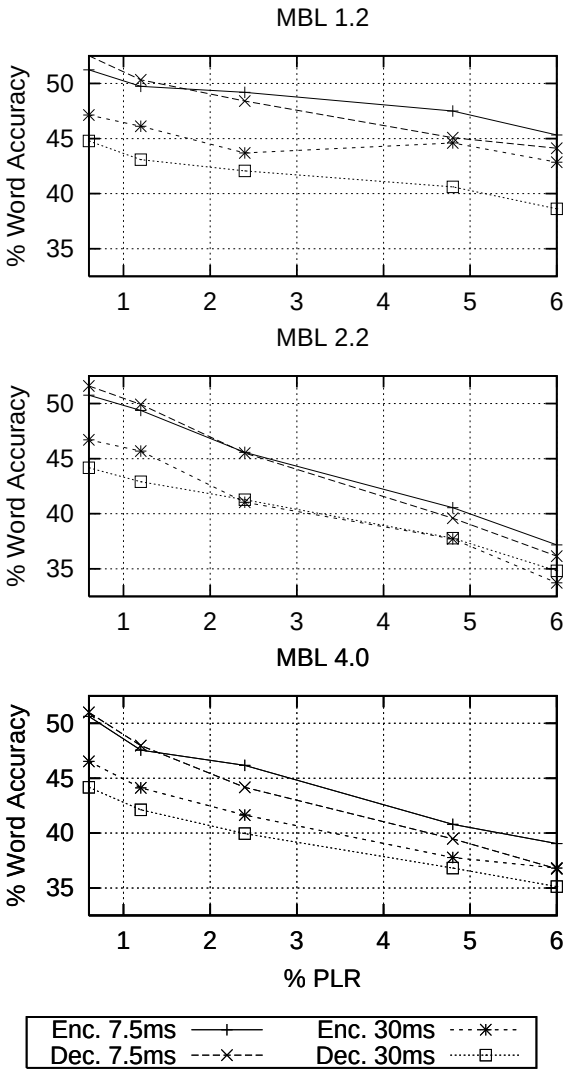

Fig. 2. *Word Accuracies, obtained on the $APASCI$ database, as functions of PLR, at different MBL values.*

5. CONCLUSIONS

We have compared different acoustic front-ends for automatic recognition of coded voice. Several experiments have been carried out on both an Italian acoustic phonetic database (*APASCI*), and on a noisy English database (*AURORA2*) of continuous digits. The chosen speech recognition tasks are purely acoustic (i.e. loop grammars have been used), so that all the results are not affected by probabilities depending upon the language model.

For both databases, we have shown that the main reason of performance drop between the baseline system, trained on original voice, and the system trained on encoded streams relies on the reduced analysis rate employed by the

codec. On the contrary, the quantization of acoustic parameters has proven to have less effects. We have also shown that at the lower frame rate ($7.5\ ms$) the contribution of the excitation signal is fundamental.

We have also measured the influence of packet loss on encoded and decoded databases, showing that it has not dramatic effects on the overall performance, unless the test set is subjected to high rates of packet loss.

To develop an efficient DSR system, working over low bandwidth networks, one has to consider that the overall bit rate is a function of both the analysis rate and the number and length of codewords that must be transmitted for each feature vector. A suitable trade off between these two last quantities has to be reached in order to reduce the bit rate, maintaining, at the same time, speech recognition performance at a sufficient level for all possible applications.

Actually, for each LSP vector of dimension 10, the codec transmits three 8-bit codewords (see [4] for the details). These are obtained by partitioning the given vector into three adjacent sub-vectors, of dimension: 3, 3 and 4 respectively, and by successively vector quantizing each of them. Hence, the total number of allocated bits for each LSP vector is 24. Nevertheless, the way of partitioning the feature vector into sub-vectors, as well as the design of the corresponding "product codebook" should be determined with the specific purpose of maximizing speech recognition performance. In fact, for speech recognition it could be more convenient to group features according to either their correlation or mutual information. Alternatively, a grouping criterion based on the minimization of the recognition error could be used. Future work will address these last topics.

6. REFERENCES

[1] C. Eccher, L. Eccher, D. Falavigna, L. Nardelli, M. Orlandi, and A. Sboner, "On the usage of automatic voice recognition in a web based medical application," in *Proc. ICASSP*, Hong-Kong, 2003, vol. 2, pp. 289–292.

[2] H. G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ISCA ITRW*, Paris, France, 2000.

[3] "Speech processing, transmission and quality aspects (stq); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms," *ETSI document ES 202 050 v1.1.1*, 2002.

[4] "Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s," *ITU-T Recommendation G723.1*, 1996.

[5] C. P. Moreno, Ã. Gallardo-Antolin, and F. Diaz de Maria, "Recognizing voice over ip: A robust front-end for speech recognition on the world wide web," *IEEE Trans. on Multimedia*, vol. 3, no. 2, pp. 209–216, 2001.

[6] B. Raj, J. Migdal, and R.Singh, "Distributed speech recognition using codec parameters," in *Proc. ASRU*, Trento, Italy, 2001, pp. 9–13.

[7] H. K. Kim and R. V. Cox, "A bitstream-based front-end for wireless speech recognition on is-136 communications system," *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 5, pp. 558–568, 2001.

[8] B. Angelini, F. Brugnara, D. Falavigna, D. Giuliani, R. Gretter, and M. Omologo, "Speaker independent continuous speech recognition using an acoustic-phonetic corpus," in *Proc. ICSLP*, Yokohama, Japan, 1994, pp. 1391–1394.

[9] L. N. Kanal and A.R.K. Sastry, "Models for channels with memory and their applications to error control," *Proc. IEEE*, vol. 66, pp. 724–744, 1978.

[10] M. S. Borella, "Measurement and interpretation of internet packet loss," *Journal on Communication Networks*, vol. 2, no. 2, pp. 93–102, 2000.