Evaluating the Utility of Media–Dependent FEC in VoIP Flows

Gerardo Rubino and Martín Varela

Irisa - INRIA/Rennes
Campus universitaire de Beaulieu
35042 Rennes CEDEX, France
{rubino|mvarela}@irisa.fr

Abstract. In this paper, we present an analysis of the impact of using media–dependent Forward Error Correction (FEC) in VoIP flows over the Internet. This error correction mechanism consists of piggy-backing a compressed copy of the contents of packet n in packet $n + i$ (i being variable), so as to mitigate the effect of network losses on the quality of the conversation. To evaluate the impact of this technique on the perceived quality, we propose a simple network model, and study different scenarios to see how the increase in load produced by FEC affects the network state. We then use a *pseudo–subjective* quality evaluation tool that we have recently developed in order to assess the effects of FEC and the affected network conditions on the quality as perceived by the end–user.

1   Introduction

In recent years, the growth of the Internet has spawned a whole new generation of networked applications, such as VoIP, videoconferencing, video on demand, music streaming, etc. which have very specific, and stringent, requirements in terms of network QoS. In this paper we will focus on VoIP technology, which has some particularities with respect to other real–time applications, and it is one of the most widely deployed to date. The current Internet infrastructure was not designed with these kinds of applications in mind, so multimedia applications' quality is very dependent on the capacity, load and topology of the networks involved, as QoS provisioning mechanisms are not widely deployed. Therefore, it becomes necessary to develop mechanisms which allow to overcome the technical deficiencies presented by current networks when dealing with real–time applications.

Voice–over–IP applications tend to be sensitive to packet losses and end–to–end delay and jitter. In this paper we will concentrate on the effect of FEC on packet loss, and the effect of both on the perceived quality. While it has been shown [1] that delay and jitter have a significant impact on the perceived quality, we will focus on one–way flows, whose quality is largely dominated (at the network level) by the packet loss process found in the network. The effects of FEC on interactive (two–way) VoIP applications is the subject of future studies.

In order to assess the variations in perceived quality due to the use of FEC, we will use a technique we have recently developed [2,3]. The idea is to train an appropriate tool (a Random Neural Network, or RNN) to behave like a "typical" human evaluating the streams. This is not done by using biological models of perception organs but by identifying an appropriate set of input variables related to the source and to the network, which affect the quality, and to teach the RNN the relationships between these variables and the perceived quality. One of the main characteristics of this approach is that the result is extremely accurate (as it matches very well the result obtained by asking a team of humans to evaluate the streams). In [4], we applied this method to analyze the behavior of audio communications on IP networks, with very good results after comparison with real human evaluations.

The rest of the paper is organized as follows. Section 2 presents the tool we used to assess the perceived quality of the flows. Section 3 presents the network model we used for our analysis, and the effects of adding FEC to the audio traffic. In Section 4, we present our analysis of the effects of FEC on the quality of the flows. Finally, Section 5 presents our conclusions.

2   Assessing the Perceived Quality

Correctly assessing the perceived quality of a speech stream is not an easy task. As quality is, in this context, a very subjective concept, the best way to evaluate it is to have real people do the assessment. There exist standard methods for conducting *subjective* quality evaluations, such as the ITU-P.800 [5] recommendation for telephony. The main problem with subjective evaluations is that they are very expensive (in terms of both time and manpower) to carry out, which makes them hard to repeat often. And, of course, they cannot be a part of an automatic process.

Given that subjective assessment is expensive and impractical, a significant research effort has been done in order to obtain similar evaluations by *objective* methods, i.e., algorithms and formulas that measure, in a certain way, the quality of a stream. The most commonly used objective measures for speech/audio are Signal-to-Noise Ratio (SNR), Segmental SNR (SNRseg), Perceptual Speech Quality Measure (PSQM) [6], Measuring Normalizing Blocks (MNB) [7], ITU E–model [8], Enhanced Modified Bark Spectral Distortion (EMBSD) [9], Perceptual Analysis Measurement System (PAMS) [10] and PSQM+ [11]. These methods have three main drawbacks: (i) they generally don't correlate well with human perception [12,8]; (ii) virtually all of them (one exception is the E-model) are *comparing techniques* between the original and the received stream (so they need the former to perform the evaluation, which precludes their use in a live, real–time networking context), and (iii) they generally don't take into account network parameters. Points (ii) and (iii) are due to the fact that they have been mainly designed for analyzing the effect of coding on the streams' quality.

The method used here [2,3] is a hybrid between subjective and objective evaluation. The idea is to have several distorted samples evaluated subjectively,

and then use the results of this evaluation to teach a RNN the relation between the parameters that cause the distortion and the perceived quality. In order for it to work, we need to consider a set of P parameters (selected *a priori*) which may have an effect on the perceived quality. For example, we can select the codec used, the packet loss rate of the network, the end–to–end delay and/or jitter, etc. Let this set be $\mathcal{P} = \{\pi_1, \dots, \pi_P\}$. Once these *quality–affecting* parameters are defined, it is necessary to choose a set of representative values for each π_i, with minimal value $\pi_{\min}$ and maximal value $\pi_{\max}$, according to the conditions under which we expect the system to work. Let $\{p_{i1}, \cdots, p_{iH_i}\}$ be this set of values, with $\pi_{\min} = p_{i1}$ and $\pi_{\max} = p_{iH_i}$. The number of values to choose for each parameter depends on the size of the chosen interval, and on the desired precision. For example, if we consider the packet loss rate as one of the parameters, and if we expect its values to range mainly from 0% to 5%, we could use 0, 1, 2, 5 and perhaps also 10% as the selected values. In this context, we call *configuration* a set with the form $\gamma = \{v_1, \dots, v_P\}$, where v_i is one of the chosen values for p_i.

The total number of possible configurations (that is, the number $\prod_{i=1}^{P} H_i$) is usually very large. For this reason, the next step is to select a subset of the possible configurations to be subjectively evaluated. This selection may be done randomly, but it is important to cover the points near the boundaries of the configuration space. It is also advisable not to use a uniform distribution, but to sample more points in the regions near the configurations which are most likely to happen during normal use. Once the configurations have been chosen, we need to generate a set of "distorted samples", that is, samples resulting from the transmission of the original media over the network under the different configurations. For this, we use a testbed, or network simulator.

Formally, we must select a set of M media samples (σ_m), $m = 1, \cdots, M$, for instance, M short pieces of audio (subjective testing standards advise to use sequences having an average 10 sec length -following [5], for instance). We also need a set of S configurations denoted by $\{\gamma_1, \cdots, \gamma_S\}$ where $\gamma_s = (v_{s1}, \cdots, v_{sP})$, v_{sp} being the value of parameter π_p in configuration γ_s. From each sample σ_i, we build a set $\{\sigma_{i1}, \cdots, \sigma_{iS}\}$ of samples that have encountered varied conditions when transmitted over the network: sequence σ_{is} is the sequence that arrived at the receiver when the sender sent σ_i through the source-network system where the P chosen parameters had the values of configuration γ_s.

Once the distorted samples are generated, a subjective test [5] is carried out on each received piece σ_{is}. After a statistical screening of the answers (to eliminate "bad" observers), the sequence σ_{is} receives the value μ_{is} (often, this is a *Mean Opinion Score*, or MOS), the average of the values given to it by the set of observers. The idea is then to associate each configuration γ_s with the value $\mu_s = (1/M) \sum_{m=1}^{M} \mu_{ms}$.

At this step we have a set of S configurations $\gamma_1, \dots, \gamma_S$, and we associate μ_s with configuration γ_s. We randomly choose S_1 configurations among the S available. These, together with their values, constitute the "Training Database". The remaining $S_2 = S - S_1$ configurations and their respective values constitute

the "Validation Database", reserved for further (and critical) use in the last step of the process.

The next step is to train a specific statistical learning tool (a RNN) to learn the mapping between configurations and values as defined by the Training Database. Assume that the selected parameters have values scaled into [0,1] and the same with quality. Once the tool has "captured" the mapping, that is, once the tool is trained, we have a function $f()$ from $[0,1]^P$ into $[0,1]$ mapping now any possible value of the (scaled) parameters into the (also scaled) quality metric. The last step is the validation phase: we compare the value given by $f()$ at the point corresponding to each configuration γ_s in the Validation Database to μ_s; if they are close enough for all of them, the RNN is validated (in Neural Network Theory, we say that the tool *generalizes well*). In fact, the results produced by the RNN are generally closer to the MOS than that of the human subjects (that is, the error is less than the average deviation between human evaluations). As the RNN generalizes well, it suffices to train it with a small (but well chosen) part of the configuration space, and it will be able to produce good assessments for any configuration in that space. The choice of the RNN as an approximator is not arbitrary. We have experimented with other tools, namely Artificial Neural Networks, and Bayesian classifiers, and found that RNN are more performant in the context considered. ANN exhibited some performance problems due to over-training, which we did not find when using RNN. As for the Bayesian classifier, we found that while it worked, it did so quite roughly, with much less precision than RNN. Besides, it is only able to provide discrete quality scores, while the NN approach allows for a finer view of the quality function.

For this study, we will use a RNN trained with results from a subjective tests campaign carried out with 17 subjects. The subjects were presented 115 sets of speech samples that had been generated using the Robust Audio Tool (RAT [13]), corresponding to different network and coding configurations. A MOS test was performed and the results screened as per [5]. About 90 of the results obtained were used to train the RNN, and the remaining ones were used to validate it. The parameters considered for our experiment are listed on table 1, and are described below.

Codec – the primary codec (16 bit linear PCM, and GSM),
FEC – the secondary codec (GSM), if any,

Table 1. Network and encoding parameters and values used

Parameter	Values
Loss rate	0% ... 15%
Mean loss burst size	1 ... 2.5
Codec	PCM Linear 16 bits, GSM
FEC	ON(GSM)/OFF
FEC offset	1 ... 3
Packetization interval	20, 40, and 80ms

FEC offset – the offset, in packets, of the redundant data (we used offsets of 1 and 3 packets for the Forward Error Correction (FEC) scheme presented in [14,15].

Packetization Interval (PI) – the length (in milliseconds) of audio contained in each packet (we considered packets containing 20, 40 and 80ms).

Packet loss rate – the percentage of lost packets (2, 7, 10 and 15%).

Mean loss burst size – the average size of packet loss bursts (1, 1.7 and 2.5 packets); we consider this parameter to have a finer view on the packet loss process than the one reduced to the packet loss rate only.

So, in order to obtain an estimation of the perceived quality, all that is needed is to feed the trained RNN with values for those parameters, and it will output a MOS estimation, very close to the actual MOS.

3   The Network Model

The tool described in the last section gives us a way to explore the perceived quality as a function of the 6 selected parameters. For instance, this allows to plot MOS against, say, packet loss rate in different cases (parameterized by the 5 remaining parameters), etc. For performance evaluation purposes, we want to know which is the impact on quality of typical *traffic* parameters (such as throughputs) and the parameters related to the *dimensions* (such as windows, buffer sizes, etc.). This gap is here bridged by adding a network model.

In this paper we will consider a very simple network model, much like the one presented in [16,17]. It consists of an $M/M/1/H$ queue which represents the bottleneck router in the network path considered. In spite of its simplicity, this model will allow us to capture the way FEC affects *perceived* quality. Moreover, it appears to be quite robust (see the comments in 3.3). We will concern ourselves with two classes of packets, namely audio packets and background traffic. Audio packets can have FEC or not, but we will consider that if FEC is on, then all flows are using it. Our router will have a drop–tail buffer policy, which is common in the current Internet.

3.1   Transmission Without FEC

First, consider the case of audio without FEC. We will take the standard environment in the $M/M/1/H$ case: Poisson arrivals, exponential services, and the usual independence assumptions. The arrival rates of class i units is λ_i pps (packets per second) and the link has a transmission capacity of c bps. The average packet length for class i packets is B_i bits. In order to be able to use analytical expressions, we consider in the model the global average length of the packets sharing the link, B, given by $B = \alpha_1 B_1 + \alpha_2 B_2$, where $\alpha_i = \lambda_i/\lambda$, with $\lambda = \lambda_1 + \lambda_2$. The service rate of the link *in pps* is then $\mu = c/B$.

Let us assume that the buffer associated with the link has capacity equal to N bits. Then, in packets, its capacity will be taken equal to $H = N/B$. To

simplify the analysis, we will use the expressions for the performance metrics of the $M/M/1/H$ models even if H is not an integer; this does not significantly change the results and the exposition is considerably more clear.

If we denote $\varrho = \lambda/\mu$, then the packet loss probability p is

$$p = \frac{1 - \varrho}{1 - \varrho^{H+1}} \varrho^H$$

(assuming $\varrho \neq 1$). We also need to compute the mean size of loss bursts for audio packets, since the correlation between losses influences the speech quality. Here, we must discuss different possible definitions for the concept of loss burst, because of the multi-class context. To this end, let us adopt the following code for the examples we will use in the discussion: a chunk of the arrival stream at the link will be denoted by "$\ldots, x, y, z, \ldots$" where symbols $x, y, z, \ldots$ are equal to i if a class-i packet arrives and is accepted (the queue was not full) and to $\bar{i}$ if the arrival is a class-i one and it is rejected (because there was no room for it).

Assume a packet of class 1 arrives at a full queue and is lost, and assume that the previous class 1 packet was not lost. Then, a burst of class 1 losses starts. Strictly speaking, the burst is composed of a set of consecutive audio packets all lost, whatever happens to the class-2 packets arriving between them. For instance, in the path "$\ldots, 1, 2, \bar{1}, \bar{1}, 2, 2, \bar{1}, \bar{2}, 2, \bar{2}, 1, \ldots$" there is a class 1 burst with loss burst size size LBS = 3. If we use this definition of burst, when audio packets are a small fraction of the total offered traffic, this definition can exaggerate the effective impact of correlation. Even in the case of many audio packets in the global arrival process, allowing the class-2 packets to merge inside audio loss bursts can be excessive. On the other extreme, we can define an audio loss burst as a consecutive set of class-1 arrivals finding the buffer full, without class-2 packets between them. In the path shown before, if we consider this burst definition, there is a first burst of class-2 losses with size 2, then a second one composed of only one packet. Consider now the piece of path "$\ldots, 1, 2, 2, \bar{1}, \bar{1}, \bar{1}, \bar{2}, \bar{2}, \bar{1}, 2, 2, 2, \bar{1}, \bar{1}, 1, \ldots$". An intermediate definition consists of considering that we accept class-2 packets inside the same audio burst only if they are also lost (because this corresponds basically to the same congestion period). In the last example, this means that we have a loss burst with size 4. We will keep this last definition for our analysis.

Let us denote by LBS the loss burst size (recall that we focus only on class 1 units). The probability that a burst has size strictly greater than n is the probability that, after a class 1 loss, the following n class-1 arrivals are losses, accepting between them class-2 losses in any number. This means

$$\Pr(\text{LBS} > n) = p^n,$$

where

$$p = \sum_{k \geq 0} \left(\frac{\lambda_2}{\lambda_1 + \lambda_2 + \mu} \right)^k \frac{\lambda_1}{\lambda_1 + \lambda_2 + \mu} = \frac{\lambda_1}{\lambda_1 + \mu}.$$

The last relationship comes from the fact that we allow any number of class 2 units to arrive as far as their are lost, between two class 1 losses (that is, while

the burst is not finished, no departure from the queue is allowed). Using the value of p, we have

$$E(\text{LBS}) = \sum_{n \geq 0} \Pr(\text{LBS} > n) = \sum_{n \geq 0} \left(\frac{\lambda_1}{\lambda_1 + \mu} \right)^n = 1 + \frac{\lambda_1}{\mu}.$$

3.2   Transmission With FEC

If FEC is used, each audio packet has supplementary data, and we denote this overhead by r. If B' is the mean audio packet length when FEC is used, then $B'_1 = B_1(1 + r)$. The rest of the parameters are computed as before. We have $B' = \alpha_1 B'_1 + \alpha_2 B_2$, $\mu' = c/B'$, $H' = N/B'$ and $\varrho' = \lambda/\mu'$. This leads to the corresponding expression for the loss probability $p' = \varrho'^{H'}(1 - \varrho')/(1 - \varrho'^{H'+1})$ and $E(\text{LBS}') = 1 + \lambda_1/\mu'$.

3.3   About the Model Robustness

The simplicity of the classical $M/M/1/H$ single class model hides its capacity a capture the dynamics of the system. We also explored the more directly multi-class $M/M/1/H$ FIFO queue where the service rate is μ_i for class-i packets. This model can be easily numerically analyzed by writing and solving the associated equilibrium equations.

Let us denote

$q_i = \Pr(\text{in steady state, the queue is saturated and a class-}i\text{ packet is being transmitted}).$

Then, using the previously definition of class-k loss bursts, their average length $E(\text{LBS}_k)$ is derived exactly as before. First, conditioned on the class of the packet being transmitted,

$$E(\text{LBS}_k \mid \text{a class-}i\text{ packet is being transmitted}) = 1 + \lambda_k/\mu_i.$$

Then, given the low loss rate considered, we average the average loss burst length for class-k as

$$E(\text{LBS}_k) = \frac{\sum_i q_i(1 + \lambda_k/\mu_i)}{\sum_j q_j}.$$

We explored the perceived quality as described before within this multi-class model and the numerical results we obtained were very similar to those presented in next section.

4   The Impact of FEC on VoIP Flows

The main idea of using FEC is that as the real–time requirements of VoIP make it impossible to retransmit lost packets, it is possible to mitigate the effect of

the losses by transmitting the information contained in each packet more than once. To this end, we send one (maybe more) copy of packet n's contents in packet $n + i$ ($n + i + 1$, and so on if several copies are sent), i being variable. The extra copies are normally compressed at a lower bit rate, so as to minimize the increase in bandwidth usage. When a packet is lost, if the packet containing the redundant copy of the lost one arrives in a timely fashion, the receiver can use said copy to recover from the loss, in a way that minimizes the perceived quality degradation. The variability of i is due to the bursty nature of packet losses on IP networks [18,19], and it allows to improve the performance of the FEC scheme described above by avoiding the loss of all copies of a given voice packet in the same loss burst. However, i should remain as close to 1 as possible, in order to minimize the increase of the end–to–end delay, which is known to degrade the quality of a two–way VoIP session.

In our study, we focus on the perceived audio quality, and so it is sufficient to consider only one–way flows. Therefore, we didn't consider the effects of FEC on the end–to–end delay.

In order to assess the impact on FEC at the network loss level, we must consider two factors:

- the amount of redundancy introduced by the use of FEC (and therefore the increase in packet size for the flows using FEC),
- the proportion of flows using FEC, which allows us to derive the new network load.

While the increase in size is likely to be small, more or less equal for different applications, and certainly bounded by a factor of 1, the amount of flows using FEC is very hard to determine. Estimations of the number of VoIP flows on the Internet are not readily available in the literature, but there are some estimations [20] that say that UDP traffic only accounts for about 15 to 20% of the total network traffic. Even then, VoIP is probably still a modest fraction of that UDP traffic (some studies [21] suggest that about 6% of traffic corresponds to streaming applications). However, being that VoIP applications have a growing user base, it seems reasonable that in some time they may account for a higher fraction of the total Internet traffic.

We studied different scenarios, with increasing volumes of VoIP traffic in order to assess the impact of FEC in the quality perceived by the end–user. For simplicity's sake, we assumed that if FEC is in use, then all VoIP flows are using it.

For our example, we'll consider an T3–type line which runs at 45Mbps (actual speeds are not really relevant, since we will concern ourselves with the load of the network – numbers are used to better set the example). In practice, buffer space in core routers is not limited by physical memory, but rather by the administrators, who may wish to minimize delays (while this implies that the loss rate is higher than it could be, it is important to make TCP flows behave properly). This time is normally limited to a few milliseconds [22]. We will use a value of 200ms (which requires a buffer space of $N = 45\text{Mbps} * 0.2\text{s} = 9\text{Mbits}$, or about 1965 packets of 600B), which is on par with current practices.

Table 2. Packet size distribution

Size (Bytes)	Probability
40	50
512	0.25
1500	0.24
9180	0.01

Some studies [23] show that an important percentage (about 60%) of packets have a size of about 44B, and that about 50% of the volume of bytes transferred is transferred on 1500B or higher packet sizes. Table 2 shows the distribution we chose for our background traffic packet sizes, which yields an average packet size B_2 of about 600B.

We considered that network load varies between 0.5 and 1.15. Smaller values have no practical interest, since in the model we used, they result in negligible loss rates, and higher load values result in loss rates above 15%, which typically yield unacceptable quality levels.

As for the fraction of VoIP traffic, we studied different values, between 5% and 50% of the total packet count. Granted, the higher values are currently unrealistic, but they may make sense in a near future if more telephony providers move their services toward IP platforms, and other VoIP applications gain more acceptance.

We chose to use PCM audio with GSM encoding for FEC, and an offset of one packet for the redundant data. This is a bit pessimistic, since sample–based codecs are not very efficient for network transmission, but it allows us to get a worst–case scenario of sorts (since increasing the fraction of audio traffic does indeed increase network load). In this case, the increase in packet size is of about 10% when using FEC, which gives payloads of 353B, against 320B of PCM–only traffic for 20ms packets. We also tried GSM/GSM, which results in a much smaller packet size (66B and 33B with and without FEC respectively), but found that the results are qualitatively similar to those of PCM, and so we will discuss only the PCM ones.

4.1   Assessing the Impact of FEC on the Perceived VoIP Quality

We present our results as a series of curves, plotting the estimated MOS values against network load, and for different values of the proportion of voice packets.

As can be seen in the curves in Figure 1, using FEC is beneficial for the perceived quality in all the conditions considered. It can be seen that when the proportion of voice traffic becomes important ($> 30\%$) the performance of the FEC protection decreases. We believe that this is related to the fact that a higher proportion of voice packets implies a higher mean loss burst size (which is still smaller than 2 in our model for the conditions considered), and being that we chose an offset of 1 packet for the FEC, it is logical to see a slight
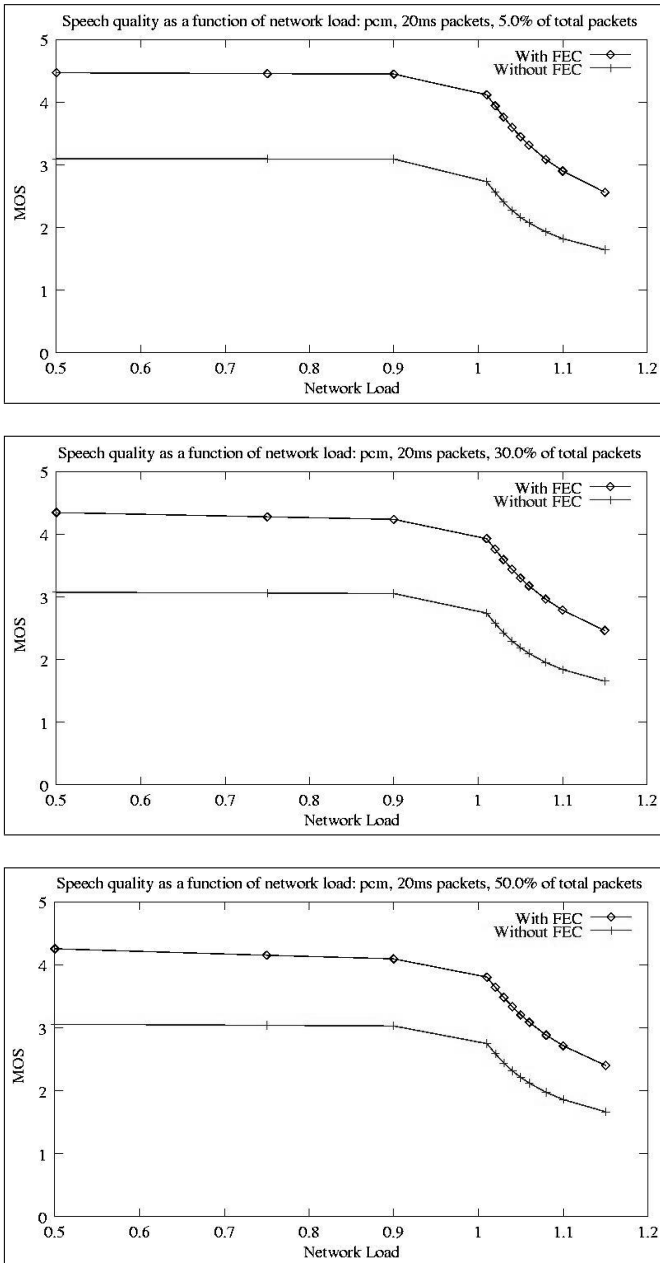
Fig. 1. MOS as a function of network load for different fractions of voice traffic, with and without FEC (20ms packets)

decrease in performance.We did not find, however, a negative impact of FEC on the perceived quality, as predicted in [16,17]. Even when most of the packets carry voice with FEC, we found that it is better to use FEC than not to use it. It might be that for extremely high values of the network load, this will not hold, but in that case quality will be be already well below acceptable values, FEC or no FEC, so it doesn't really matter.

The second strong fact coming from these numerical values is that the qualitative behaviour of perceived quality when load increases is basically the same in all cases, with a significant degradation when load approaches 1 and beyond.

We also found that using a larger larger packetization interval can help improve the perceived quality. We tried doubling the packet size, obtaining 40ms packets (20ms and 40ms packets are commonly used in telephony applications), and obtained a slight increase in quality even for higher proportions of audio packets (25% of audio packets, which corresponds to the same number of flows of a 50%–20ms audio packet proportion). This can be seen in Figure 2. While increasing the packetization interval is beneficial in one–way streams, it should be studied whether the higher delay values that this creates do not counter these benefits.
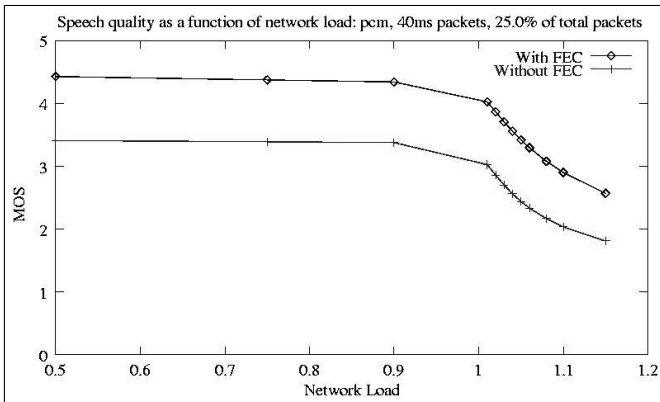

Fig. 2. MOS as a function of network load for 25% of voice traffic, with and without FEC (40ms packets)

5   Conclusions

In this paper we analyze the effect of media–dependent FEC on one–way speech streams. To this end, we studied the effects of the increase in network load generated by adding FEC to voice streams, using a simple queueing model to represent the bottleneck router, in a similar fashion as in[16,17].

In order to estimate the voice quality perceived by the end–user, we used a method we have recently proposed, based on Random Neural Networks trained

with data obtained from subjective quality assessments. We fed the RNN with coding parameters and with packet loss parameters derived from the network model, and obtained MOS estimates for each configuration.

We considered a range of network loads that yielded reasonable loss rates, and several values for the fraction of packets corresponding to speech flows. The results obtained indicate that, as expected, the use of FEC is always beneficial to the perceived quality, provided that the network parameters stay within reasonable ranges. Our approach allows to provide actual quantitative estimates of this gain, as a function of different parameters. In the paper we focused on the network load, but similar analysis can be done to explore many other interesting aspects, such as delay/jitter for interactive flows.

One of the strong points of this approach is the coupling of an accurate technique to assess perceived quality (avoiding the use of abstract "utility functions") with a model of the network allowing to obtain information about the loss process. If for some reason the model used yesterday is not considered appropriate today, one can move to some other (possibly more detailed) representation of the bottleneck (or perhaps to a tandem queueing network corresponding to the whole path followed by the packets), and use the same approach to finally map traffic parameters to final, end–to–end quality *as perceived by the final user*.

References

1. Claypool, M., Tanner, J.: The effects of jitter on the perceptual quality of video. In: Proceedings of ACM Multimedia Conference. (1999)
2. Mohamed, S., Cervantes, F., Afifi, H.: Integrating networks measurements and speech quality subjective scores for control purposes. In: Proceedings of IEEE INFOCOM'01, Anchorage, AK, USA (2001) 641–649
3. Mohamed, S., Rubino, G.: A study of real–time packet video quality using random neural networks. IEEE Transactions On Circuits and Systems for Video Technology **12** (2002) 1071 –1083
4. Mohamed, S., Rubino, G., Varela, M.: Performance evaluation of real-time speech through a packet network: a random neural networks-based approach. Performance Evaluation **57** (2004) 141–162
5. ITU-T Recommendation P.800: (Methods for subjective determination of transmission quality)
6. Beerends, J., Stemerdink, J.: A perceptual speech quality measure based on a psychoacoustic sound representation. Journal of Audio Eng. Soc. **42** (1994) 115–123
7. Voran, S.: Estimation of perceived speech quality using measuring normalizing blocks. In: IEEE Workshop on Speech Coding For Telecommunications Proceeding, Pocono Manor, PA, USA (1997) 83–84
8. ITU-T Recommendation G.107: (The E-model, a computational model for use in transmission planning)
9. Yang, W.: Enhanced Modified Bark Spectral Distortion (EMBSD): an Objective Speech Quality Measrure Based on Audible Distortion and Cognition Model. PhD thesis, Temple University Graduate Board (1999)

10. Rix, A.: Advances in objective quality assessment of speech over analogue and packet-based networks. In: the IEEE Data Compression Colloquium, London, UK (1999)
11. Beerends, J.: Improvement of the p.861 perceptual speech quality measure. ITU-T SG12 COM-34E (1997)
12. Hall, T.A.: Objective speech quality measures for Internet telephony. In: Voice over IP (VoIP) Technology, Proceedings of SPIE. Volume 4522., Denver, CO, USA (2001) 128–136
13. London, U.C.: Robust Audio Tool website. (`http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/index.html`)
14. IETF Network Working Group: RTP payload for redundant audio data (RFC 2198) (1997)
15. Bolot, J.C., Fosse-Parisis, S., Towsley, D.: Adaptive FEC–based error control for Internet telephony. In: Proceedings of INFOCOM '99, New York, NY, USA (1999) 1453–1460
16. Altman, E., Barakat, C., R., V.M.R.: On the utility of FEC mechanisms for audio applications. Lecture Notes in Computer Science **2156** (2001)
17. Altman, E., Barakat, C., R., V.M.R.: Queueing analysis of simple FEC schemes for IP telephony. In: Proceedings of INFOCOM '01. (2001) 796–804
18. Yajnik, M., Moon, S., Kurose, J., Towsley, D.: Measurement and modeling of the temporal dependence in packet loss. In: Proccedings of IEEE INFOCOM '99. (1999) 345–352
19. Hands, D., Wilkins, M.: A study of the impact of network loss and burst size on video streaming quality and acceptability. In: Interactive Distributed Multimedia Systems and Telecommunication Services Workshop. (1999)
20. Fomenkov, M., Keys, K., Moore, D., Claffy, K.: Longitudinal study of Internet traffic in 1998-2003. `http://www.caida.org/outreach/papers/2003/nlanr/nlanr_overview.pdf` (2003) CAIDA, San Diego Super Computing Center, University of California San Diego.
21. Fraleigh, C., Moon, S., Lyles, B., Cotton, C., Kahn, M., Moll, D., Rockell, R., Seely, T., Diot, C.: Packet-level traffic measurements from the sprint ip backbone. IEEE Network **17** (2003) 6–17
22. Various Authors: Discussion on the e2e mailing list: "Queue size for routers?". (`http://www.postel.org/pipermail/end2end-interest/2003-January/002643.html`)
23. Claffy, K., Miller, G., Thompson, K.: The nature of the beast: Recent traffic measurements from an Internet backbone. In: INET '98, Geneva, Switzerland (1998)