On the utility of FEC mechanisms for audio applications

Eitan Altman[1,2], Chadi Barakat[1], and Víctor M. Ramos R.[3] ⋆

[1] INRIA Sophia Antipolis, France
{altman, cbarakat}@sophia.inria.fr
[2] CESIMO, Universidad de Los Andes, Venezuela
[3] Institut Eurécom
06904 Sophia Antipolis, France
Victor.Ramos@eurecom.fr

Abstract. FEC mechanisms have been proposed to recover from packet losses, and hence to improve the perceived quality in audio applications. Recently, it has been shown in [1] that the redundancy added by a FEC scheme increases the congestion of the network and deteriorates the audio quality instead of improving it. In this work we show via a simple queuing analysis that the impact of FEC on the audio quality is not always negative and that we can get better quality in some scenarios. In particular, we show that FEC is beneficial when a small number of flows implement it or when the audio applications have some particular utility functions. We derive conditions on when to get a gain in quality as well as bounds on the maximum gain that we can obtain.

1 Introduction

Forward Error Correction (FEC) is now considered as the most appropriate solution for the recovery from packet losses in audio (or more generally multimedia) applications [2]. This technique consists in transmitting, together with the original audio packets, some redundant information that can be used at the audio receiver to reconstruct any packet lost within the network. Generally, the reconstruction of the lost audio packets should improve the intelligibility of the received audio signal. The redundant information is constructed at the audio source using the original packets and it is sent to the destination in separate packets or piggybacked in subsequent ones. The main advantage of FEC, which makes it very suitable for audio applications, is that packet losses can be reconstructed on runtime without any retransmission from the side of the source. This runtime reconstruction reduces the variations of the end-to-end delay since the receiver is no longer needed to wait until the source retransmits the lost packets. The variation on the end–to–end delay, often called *jitter*, is an important factor in assessing the quality of an audio transmission. Audio receivers need to implement playout buffers in order to absorb these variations and play audio packets at a regular rate [3]. An important jitter will then result in an important buffering time, in an important end-to-end delay, and hence in a poor quality[1].

⋆ The author is also an associate professor at the Universidad Autónoma Metropolitana in Mexico City.

[1] An audio conversation is considered to be *interactive* if the two-way end-to-end delay is less than 250ms, including media coding and decoding, network transit and playout buffering [4].

This advantage of FEC has motivated many developers of audio applications to incorporate it into their tools (e.g., Freephone [5] and Rat [6]). Different FEC schemes have been proposed in the literature for this purpose: parity and block erasure codes, convolutional codes, interleaving, multiple description codes, etc. We will focus in this paper on a simple FEC scheme that has been standardized [7] by the IETF (Internet Engineering Task Force) and that has been implemented in many recent audio tools. The scheme simply consists in adding a redundant copy of the original audio packet to the tail of the subsequent one. If it happens that an audio packet is lost while crossing the network and that the following packet is correctly received, the lost packet can be reconstructed from the redundant information contained in the following one. Figure 1 depicts a particular case of this simple FEC scheme, where the offset (ϕ) between the original packet and its copy is equal to 1. Usually, the redundant information is obtained by coding the original packets with a low bit-rate codec. For example, an original audio packet can be coded with PCM and its copy with GSM [8] or LPC [9].

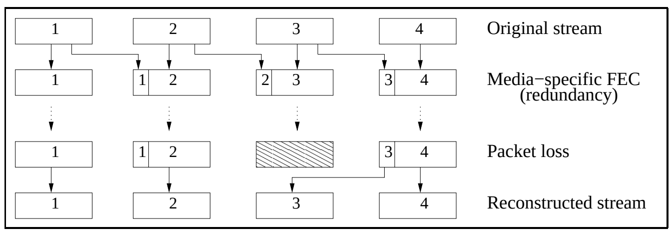


Fig. 1. Simple FEC mechanism where packet $n + 1$ carries redundant information on packet n.

Different works have tried to improve the performance of this simple FEC scheme. Some authors propose to increase the offset (ϕ) between the original packet and its copy [10, 11]. Their argument is that the loss process of packets in the Internet is bursty [12–16] and hence, by moving away the redundancy from the original packet, we increase the probability that the redundant copy of an audio packet is correctly received when the original packet is lost. Other authors propose to add multiple redundant copies of a packet in multiple subsequent ones [10, 17]. The authors in [18] show that, by adding to an audio packet a redundant copy computed from a block of some preceding packets, we get a better audio quality than when adding to the audio packet a redundant copy computed from a single preceding one. In [18], the authors proposed different ways to group packets in blocks. But, all these works ignore an important fact, that the addition of redundancy increases the transmission rate of the audio sources which may increase the load of the network and hence the loss probability of packets. The study of the performance of a FEC scheme under a constant loss rate leads certainly to an improvement in quality since the number of packets played at the receiver is larger. However, the quality may deteriorate instead of improving if the loss rate of audio packets considerably increases due to the addition of FEC. In this later case, the addition of FEC will not compensate the increase in the loss rate caused by FEC.

Recently, it has been shown in [1] via a queuing analysis that the simple FEC scheme we outlined above does not lead to an improvement of audio quality. The authors in [1] considered a single bottleneck node for the network and focused on the case when the buffer size in the bottleneck router is only dedicated to the audio flow (or to an aggregate of audio flows implementing the same FEC scheme and sharing the same bottleneck). The assumptions made in [1] hold when all flows in the network implement FEC, or when a round-robin scheduler with per-flow queuing is used. Under these assumptions, the authors in [1] show that even for the infinite-offset case ($\phi \to \infty$) which forms an upper bound on the audio quality, adding FEC according to this simple FEC scheme leads always to a deterioration of quality caused by an important increase in network load.

In this work we address the questions of how and where this simple FEC scheme, which we recall is implemented in many audio tools as Freephone and Rat, leads to an improvement in quality. The negative result given in [1] holds in the case when all the flows in the network add FEC, or when the audio flow has its own buffer in network routers. It also holds with the particular utility function the authors considered. A *utility function* indicates the variation of the audio quality at the receiver as a function of the transmission rate. The authors in [1] considered a linear utility function; they supposed that the more the user receives data, the better is the quality and that the increase in quality for a certain amount of redundancy is the same for any value of the transmission rate. In fact, the quality of an audio transmission is quite a subjective measure and it is known to be non-linear [19]. Moreover, the audio source may use different code rates for FEC which will result in different qualities for the same value of the transmission rate. We look here at cases where the assumptions in [1] are not satisfied and we try to understand why this simple FEC scheme improves the audio quality in some scenarios. We use some queuing models for this purpose. Our findings in this paper can be summarized as follows:

– With a linear utility function as the one used in [1], the addition of FEC leads to an improvement in quality if the (total) rate of the flow(s) adding FEC is small compared to the total rate of the other flows sharing the same bottleneck and not adding FEC. The addition of FEC in this case does not lead to an important increase in the loss rate which explains this improvement. We start to lose in quality when the (total) rate of the flow(s) using FEC increases.
– The audio quality is always an increasing function of the offset between the original packet and its copy.
– In the case when all flows are adding FEC, which forms the worst case where the addition of FEC has the biggest impact on the load of the network, it is possible to obtain a gain in quality for some particular utility functions. The utility function must increase with the amount of FEC faster than the linear one, and higher increase rates are required for small amounts of FEC. In some words, to gain in quality, a small amount of FEC must lead to approximately the same quality as the original audio packet.

The remainder of this paper is organized as follows. In Section 2 we investigate the case of a single audio flow sharing the bottleneck with an exogenous traffic not using FEC. In Section 3 we study the performance of the FEC scheme described above for different utility functions. We conclude this work in Section 4. Note that although we are focusing on audio flows, our results on FEC are valid for any other kind of multimedia application.

2 Multiplexing and FEC performance

2.1 The model

Consider the case of an audio flow implementing FEC and sharing a bottleneck router with some other flows not implementing FEC. We look at the other flows as a single exogenous flow of constant rate and of packet size exponentially distributed. The latter choice can be justified by the mixture of a large number of flows from different sources and of different packet sizes. Let $1/\mu$ denote the average transmission time at the bottleneck of a packet from the exogenous flow. This time is independent of the amount of FEC added to the audio flow. We consider that the original audio packets (before the addition of FEC) have a fixed length and we denote by $1/\mu_0$ their average transmission time at the output interface of the bottleneck router.

Let us suppose that packets (audio + exogenous) arrive at the bottleneck router according to a Poisson process of constant rate λ. Suppose also that audio packets arrive at the bottleneck according to a Poisson process. This latter assumption can be justified by the fact that audio packets cross multiple routers before arriving at the bottleneck, so that their inter-arrival times can be approximated by an exponential distribution. Let $\beta \in [0,1]$ denote the fraction of arriving packets belonging to the audio flow; this quantity represents the probability that a packet arriving at the bottleneck is of audio type. Suppose finally that the bottleneck router implements the classical Drop Tail policy and has a buffer of size K packets (packet in service included). Packets from different flows share the K places of the buffer and are served in a FIFO (First-In First-Out) fashion. The system can be then considered as an $M/G/1/K$ queuing system where packets arrive according to a Poisson process and where service times (or transmission times in our settings) are independent and identically distributed. This system can be then solved using some known results from queuing theory [20, 21]. Our main objective is to find an expression for the audio quality at the destination as a function of the different system parameters as well as the amount of FEC added to the original packets by the audio source.

2.2 The analysis

Suppose first that the audio flow does not implement FEC. We look at the audio quality at the moments at which packets would arrive at the destination. We take a value equal to 1 as the quality obtained when the audio packet is correctly received, and 0 as the quality when the packet is lost in the network. The *average audio quality* during the conversation is equal to $Q = 1 - \pi$, where π denote the stationary probability that a packet is dropped in an $M/G/1/K$ system. This probability is equal to $\pi = \frac{1+(\rho-1)f}{1+\rho f}$, where ρ is the total system load (or the total traffic intensity) given by $\rho = \lambda(\frac{\beta}{\mu_0} + \frac{1-\beta}{\mu})$, and f is the $K-2$ th coefficient of the Taylor series of a complex function $G(s)$ defined as $G(s) = (B^*(\lambda(1-s))-s)^{-1}$. $B^*(s)$ is the Laplace Stieltjes transform of the service time distribution [21]. In our case,

$$B^*(s) = \int_0^\infty b(t)e^{-st}dt = \beta e^{-s/\mu_0} + (1-\beta)\mu/(\mu+s), \quad \text{for } Re(s) \geq 0\,.$$

The coefficient f can be computed by developing the Taylor series of the function $G(s)$ with some mathematical symbolic software [2]. It can also be calculated using the theorem of residues as follows:

$$f = \frac{1}{(K-2)!}\,\frac{d^{K-2}G(s)}{ds^{K-2}}\bigg|_{s=0} = \frac{1}{2\pi i}\oint_{D_r} G(s)\frac{ds}{s^{K-1}}\,,$$

where D_r is any circle in the complex plane with center 0 and with radius chosen small enough so that the circle does not contain any pole of the function $G(s)$.

Now, the addition of FEC to the audio flow according to the FEC scheme we described in Section 1 increases the transmission time of audio packets at the output interface of the bottleneck router. This increases the load of the system which changes the stationary probabilities. Let $\alpha \in [0,1]$ denote the ratio of the volume of FEC at the tail of a packet and the volume of the original packet. The new transmission time of audio packets becomes $\frac{(1+\alpha)}{\mu_0}$, and the new system load becomes

$$\rho_\alpha = \lambda\Big(\frac{\beta(1+\alpha)}{\mu_0} + \frac{(1-\beta)}{\mu}\Big)\,. \qquad (1)$$

In the same way we can compute the new transform of the transmission time, the new coefficient f, and the new drop probability of an audio packet (it is the same for exogenous packets given that the arrival processes of both flows are Poisson). Henceforth, when we add an index α to a function, we mean the new value of the function after the addition of an amount α of FEC. The quality after the addition of FEC becomes

$$Q_\alpha^\phi = (1-\pi_\alpha) + U(\alpha)\pi_\alpha(1-\pi_\alpha^\phi)\,. \qquad (2)$$

The first term corresponds to the quality obtained when the original audio packet is correctly received. The second term corresponds to the quality obtained when the redundant copy is correctly received and the original packet is lost. $U(\alpha)$ indicates how much quality we get from an amount α of FEC. The quantity π_α^ϕ indicates the probability that the packet carrying the redundancy is dropped given that the original packet is also dropped. ϕ represents the offset (in number of audio packets) between the original packet and the one containing its copy. In this section we will only consider the case of a utility function $U(\alpha) = \alpha$ similar to the one studied in [1]. We keep the study of the impact of other utility functions until Section 3.

[2] As Maple (http://www.maplesoft.com) or Mathematica (http://www.wolfram.com).

The exact computation of Q_α^ϕ requires the computation of π_α^ϕ. This latter function is quite difficult to calculate given the multiplexing of packets from both flows at the bottleneck. We must summarize over all the possible numbers of non-audio packets inserted between audio packets. What we can do instead is to find bounds on this probability and thus bounds on the quality. From [1], the probability that a packet is lost given that the n-th previous packet is lost is a decreasing function of n and it converges to π_α when $n \to \infty$. We can write $\pi_\alpha \leq \pi_\alpha^\phi \leq \pi_\alpha^0$, with π_α^0 being the probability that a packet (from any flow) is lost given that the previous packet is also lost. This gives us the following two bounds on the quality: $Q_\alpha^0 \leq Q_\alpha^\phi \leq Q_\alpha$, where

$$Q_\alpha^0 = (1 - \pi_\alpha) + \alpha\pi_\alpha(1 - \pi_\alpha^0)\,, \tag{3}$$

$$Q_\alpha = (1 - \pi_\alpha)(1 + \alpha\pi_\alpha)\,. \tag{4}$$

We use these two bounds to study how the audio quality varies for different amounts of FEC and for different intensities of audio traffic. We are sure that if we gain in Q_α^0 (lose in Q_α), we will gain (lose) in quality for any offset. Our main objective here is to show how the quality varies with FEC for different values of β. It has been shown in [1] that we always lose in quality for $\beta = 1$ (i.e., when the audio flow occupies 100% of the bandwidth at the bottleneck). All that we still need to do is to find the expression for the lower bound on the quality which can be found from the expression of π_α^0.

Theorem 1 *π_α^0 is given by $1 + \frac{B_\alpha^*(\lambda) - 1}{\rho_\alpha}$, with*

$$B_\alpha^*(\lambda) = \beta e^{-\lambda(1+\alpha)/\mu_0} + (1 - \beta)\mu/(\mu + \lambda)\,,$$

and ρ_α given by equation (1).

Proof: Consider a general $M/G/1/K$ queuing system. We have to compute the probability that a packet (say 1) is dropped given that the previous packet (say 0) is also dropped. Let $a(t) = \lambda e^{-\lambda t}$ be the distribution of time intervals between arrivals (of packets from both flows), and let $b(t)$ be the distribution of service times. Let $r(t)$ be the distribution of the residual time for the packet in service when packet 0 arrives (there is certainly a packet in service since packet 0 is supposed to be dropped). Using the results in [20], we write $r(t) = \frac{1 - B(t)}{\sigma}$. $B(t)$ is the cumulative distribution function of the service time and σ is the average service time. In our case,

$$B(t) = \beta 1\{t \geq (1 + \alpha)/\mu_0\} + (1 - \beta)(1 - e^{-\mu t}),$$

and $\sigma = \rho_\alpha/\lambda$. The probability π_α^0 is no other than

$$\pi_\alpha^0 = \int_0^\infty \frac{1 - B(t)}{\sigma}(1 - e^{-\lambda t})dt\,.$$

This is the probability that the inter-arrival time between packet 0 and packet 1 is less than the residual time of the packet in service, and we summarize over all the possible values of the residual service time. With a simple calculation on this expression and by using the new values of the load intensity and the Laplace Stieltjes Transform of the service time distribution after the addition of FEC, we can prove the theorem.

2.3 Numerical results

We solve numerically the model for the two bounds on the audio quality (Eq. 3 and 4). We set K=10 packets and λ=10000 packets/s. Without loss of generality, we set μ_0=μ. We consider four values of ρ: 0.5, 0.8, 1, and 1.5. For every value of ρ, we plot the audio quality as a function of β and α. Recall that β is the fraction of audio packets and α is the amount of FEC. Figure 2 shows the results.

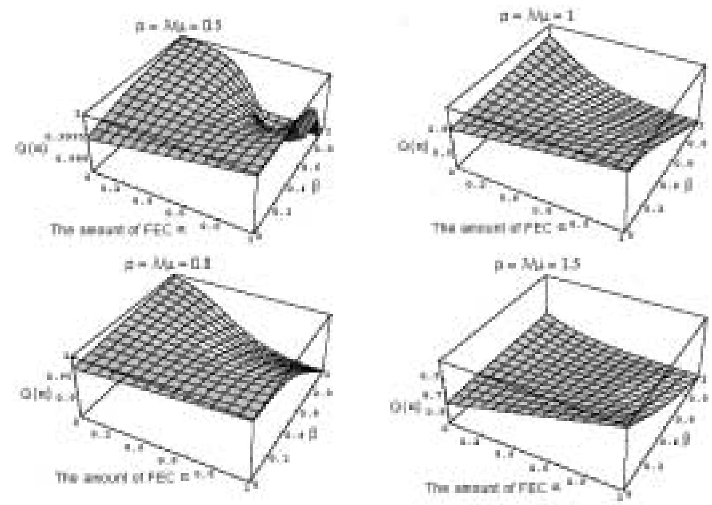


Fig. 2. Audio quality for an $M/G/1/K$ queue with two flows: the audio flow and the exogenous flow. β represents the probability that an arriving packet belongs to the audio flow. We see clearly how when $\beta \to 0$, Q_{α}^{ϕ} starts having an increasing behavior, and this gain becomes more important as ρ increases.

We conclude from the above figures that it is possible to obtain a gain with the simple FEC scheme we are studying. This requires that the intensity of the audio flow is small compared to the intensity of the other flows not implementing FEC. The gain diminishes as long as the intensity of the flows implementing FEC increases. It disappears when most of the flows start to implement FEC. This means that a FEC scheme with a simple linear utility function is not a viable mechanism. The gain that we may obtain in some cases is the result of the fact that the exogenous flows are not adding FEC and then they are not so aggressive as audio flows.

3 Utility functions and FEC performance

We seek now for a FEC mechanism able to improve the quality in the worst case when all flows in the network implement FEC. Suppose that the audio flow (or an aggregate of audio flows) uses alone the bottleneck resources ($\beta = 1$). The negative results obtained in [1] are due to the linear utility function adopted in the analysis. Adding an amount of FEC α increases the drop probability of an audio packet, which reduces the first term in the right-hand side of (2) more than it increases the second term. To get a gain, the second term must increase faster than the decrease in the first term. This can be achieved if the utility function increases faster than linearly as a function of α.

Indeed, it has been shown in [19] that multimedia applications have different utility functions than a simple linear one. These functions are typically non-linear. They are convex around zero and concave after a certain rate (between 0 and 1, with 1 being the rate that gives a utility function equal to one). Multimedia applications, and audio applications in particular, have strong delay constraints so that the quality deteriorates sharply when the transmission rate falls below a certain value. This kind of utility functions can be very useful for FEC mechanisms since the reconstruction of a packet from a copy of volume $\alpha < 1$ may give approximately the same quality as when the original packet is correctly received. We obtain a gain in quality when the redundant information we add to the original packet is small so that it does not contribute to a big increase in loss probability π, and at the same time, if reconstructed in case of the loss of the original packet, it gives a quality close to 1. Such behavior can be also obtained by coding FEC with a lower-rate codec as GSM [8]. Analytically speaking, a utility function leads to an improvement of quality if for $\alpha < 1$, we have

$$Q_\alpha^\phi = (1 - \pi_\alpha) + U(\alpha)\pi_\alpha(1 - \pi_\alpha^\phi) > (1 - \pi)\,,$$

with π being the stationary drop probability before the addition of FEC.

3.1 Some bounds on quality improvement

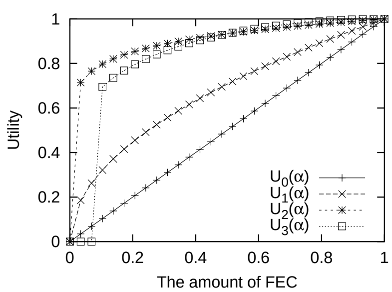
Again, we use here the bounds on the quality $Q_\alpha^0 \leq Q_\alpha^\phi \leq Q_\alpha$, with

$$Q_\alpha^0 = (1 - \pi_\alpha) + U(\alpha)\pi_\alpha(1 - \pi_\alpha^0)\,,$$
$$Q_\alpha = (1 - \pi_\alpha)(1 + U(\alpha)\pi_\alpha)$$

A utility function that improves the lower bound improves the quality for any value of ϕ. A utility function that does not improve the upper bound will not lead to an improvement of quality whatever is the value of ϕ. Using the upper bound, we can find the maximum quality that this simple FEC scheme can give and this is for the best utility function. Indeed, the best utility function is one that jumps directly to one just after 0. This could be subjectively justified by using redundant packets coded at very small rates, as LPC or GSM. A very small amount of FEC ($\alpha \simeq 0$) that does not change the load of the network (i.e., that does not change π), will then lead to the same quality as the original audio packet. The question that one may ask here is: "*why to send large original packets in this case, given that we are able to obtain the same quality with small packets?*" The important processing time required by low-rate codes could be the answer to this question. We are not addressing this issue here, and we will only focus on the calculation of an upper bound for the FEC scheme we are studying. Let Q^{max} be the maximum quality that we could obtain, thus $Q^{max} \simeq (1 - \pi) + \pi(1 - \pi) = 1 - \pi^2$.

This Q^{max} has to be compared to the quality $(1 - \pi)$ we get in the absence of FEC. Given that Q^{max} is larger than $(1 - \pi)$, we conclude that we can always find a utility function and an offset between original packets and redundancies so as to gain in quality. Note that we are not considering the impact of the coding and decoding delays on the audio quality. The impact of these delays will be the subject of a future work. We also conclude from our analysis here that the FEC scheme we are studying cannot improve the quality by more than a factor of π. This means that the maximum gain in quality we could obtain is 100% and this gain is an increasing function of the network load. For example, for a network that drops 1% of packets, we cannot improve the quality by more than 1%, and for a network that drops 10% of packets we can get an improvement up to 10%.

Without loss of generality, we consider the family of utility functions that jump from zero to 1 at a value α_0. We denote such functions by $U_{\alpha_0}(\alpha)$. These are the utility functions of the so called hard real-time applications. We also consider the upper bound on the quality (an infinite offset). When increasing the amount of FEC with such applications from 0 to α_0, the quality deteriorates since its equal to $(1-\pi_\alpha)$. When we cross α_0, the quality jumps from $(1-\pi_{\alpha_0})$ to $(1-\pi^2_{\alpha_0})$ and it resumes then its decrease with α. For such applications, the FEC scheme improves the quality if $\pi^2_{\alpha_0} < \pi$ and the maximum gain that we could obtain is a factor of $\frac{(\pi-\pi_{\alpha_0})}{(1-\pi)}$. This maximum gain corresponds to an amount of FEC slightly larger than α_0. It is not clear how the gain varies as a function of network load. But, what we can say here is that the FEC scheme behaves better with functions having a small α_0. After a certain threshold on α_0, the above condition becomes unsatisfied and it becomes impossible to gain in quality.



$$U_0(\alpha) = \alpha$$
$$U_1(\alpha) = \sqrt{\alpha}$$
$$U_2(\alpha) = \alpha^{\frac{1}{10}}$$
$$U_3(\alpha) = u(\alpha-\alpha_0)\left(\frac{1-\cos\pi\alpha}{2}\right)^{\frac{1}{10}}$$

Fig. 3. Possible utility functions for rate adaptive applications.

3.2 Some numerical results

We give in Figure 3 some possible utility functions[3] that could serve to our needs, and that are similar in their form to the utility functions proposed in [19]. In Figure 3 $U_3(\alpha)$ is plotted with $\alpha_0 = 0.1$.

We solve the model numerically for the two bounds on the quality. We calculate first the stationary distribution of the model for different values of α and ρ. We set K to

[3] The function $u(\alpha)$ is the step unit function. It is equal to 1 if $\alpha > 0$, and is equal to zero otherwise. α_0 represents the initial value giving a significant quality.

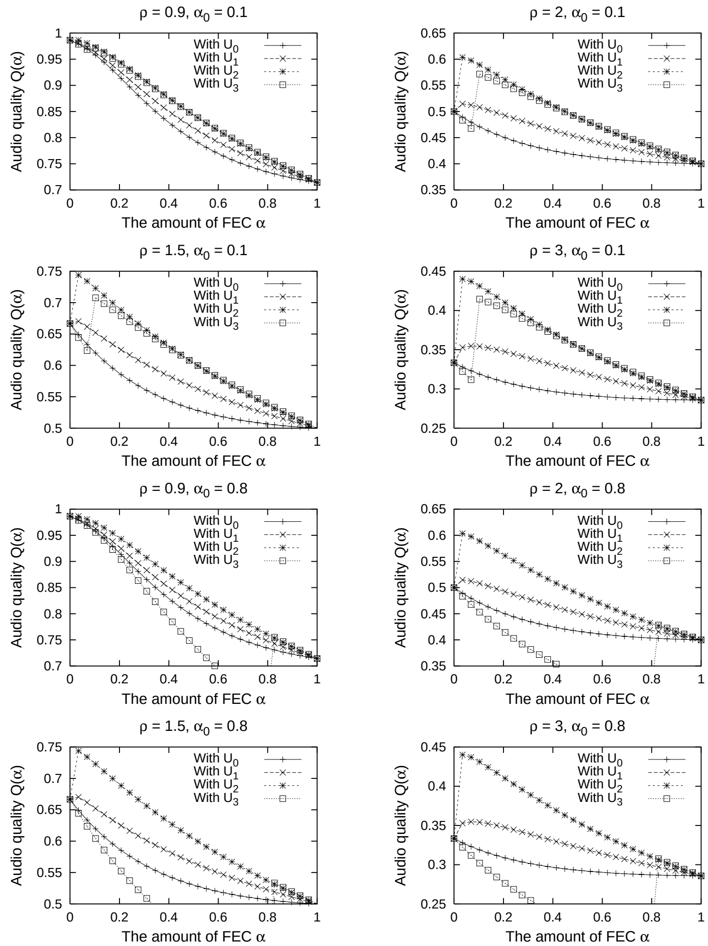


Fig. 4. Lower bound for audio quality with $K = 20$, $\alpha_0 = 0.1$ (top) and $\alpha_0 = 0.8$ (bottom).

20 and λ to 10000 packets/sec. Then, for the the different utility functions in Figure 3, we plot the upper and lower bounds on the quality (Q_{α} and Q^0_{α}). Figure 4 shows plots for the lower bound and Figure 5 shows plots for the upper bound. The top four plots were obtained with $\alpha_0 = 0.1$ and the four bottom plots with $\alpha_0 = 0.8$ in both figures. We see clearly how the jump in the utility function results in a jump in quality and how this jump leads sometimes to better quality than that at α_0 and sometimes not. We also see how the case $U(\alpha) = \alpha$ does not present any improvement in quality.

4 Conclusions

We showed in this paper that a simple FEC scheme as the one proposed by the IETF and implemented in some audio tools may lead to better performance in two cases. The

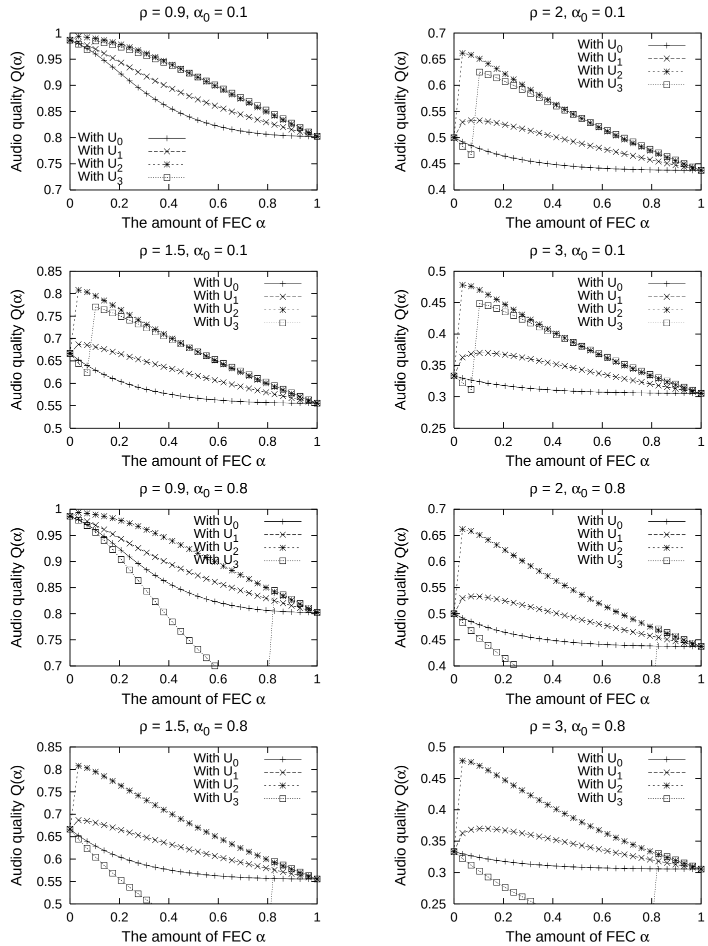


Fig. 5. Upper bound for audio quality with $K = 20$, $\alpha_0 = 0.1$ (top) and $\alpha_0 = 0.8$ (bottom).

first case is when the audio flow has a small rate compared to the exogenous traffic. The second case is when the utility function of the audio application presents an important jump at small transmission rates. We gave conditions on where the FEC scheme can improve the audio quality.

Although we found some regions where the FEC scheme can behave well, we believe that this scheme is not the appropriate solution for improving the quality of audio applications. In the current Internet, this scheme is profiting from the fact that most of the other flows are not implementing FEC. This will not be the case when all flows start to add FEC to their packets. There is also a problem with the mechanism in case of applications with different utility functions than linear. We found that we get a gain when a small amount of redundancy gives the same performance as the big original packet. It seems intuitive here to reduce the volume of original packets to reduce the drop probability and to gain in quality instead of adding FEC that does not improve the performance by no more than 100%. There is no need to send long packets if we are able to get good quality with small ones.

We believe that the main problem with this kind of mechanisms is that the redundant information is constructed at the source using one packet and so the destination has only two choices: either receive the original packet or receive its copy. Better performance could be obtained if we give the receiver more choices by constructing at the source the redundancy carried by a packet from a block of audio packets. This is what we will investigate in the future.

References

1. Altman, E., Barakat, C., Ramos, V.M.: Queueing analysis of simple FEC schemes for IP telephony. Proc. IEEE Infocom (2001)
2. Perkins, C., Hodson, O., Hardman, V.: A survey of packet loss recovery for streaming audio. IEEE Network (1998)
3. Rosenberg, J., Qiu, L., Schulzrinne, H.: Integrating packet FEC into adaptive voice playout buffer algorithms on the Internet. Proc. IEEE INFOCOM (2000)
4. Perkins, C., Hodson, O.: Options for repair streaming media. RFC 2354 (1998)
5. García, A.V., Fosse-Parisis, S.: (FreePhone audio tool) High-Speed Networking Group, INRIA Sophia Antipolis.
6. Mice Project, T.: (RAT: Robust Audio Tool) Multimedia Integrated Conferencing for European Researchers, University College London.
7. Perkins, C., Kouvelas, I., Hodson, O., Hardman, V.: RTP payload for redundant audio data. RFC 2198 (1997)
8. Scourias, J.: (Overview of the global system for mobile communications) Univ. of Waterloo.
9. Tremain, T.E.: The government standard linear predictive coding algorithm: Lpc-10. Speech Technology **1** (1982) 40–49
10. Bolot, J.C., Fosse-Parisis, S., Towsley, D.: Adaptive FEC-based error control for interactive audio in the Internet. Proc. IEEE Infocom (1999)
11. Kouvelas, I., Hodson, O., Hardman, V., Crowcroft, J.: Redundancy control in real-time Internet audio conferencing. Proc. of AVSPN (1997)
12. Altman, E., Jean-Marie, A.: Loss probabilities for messages with redundant packets feeding a finite buffer. IEEE Journal of Selected Areas in Communications **16** (1998) 779–787
13. Cidon, I., Khamisy, A., Sidi, M.: Analysis of packet loss process in high-speed networks. IEEE Transactions on Information Theory **IT-39** (1993) 98–108
14. Hellal, O.A., Altman, E., Jean-Marie, A., Kurkova, I.: On loss probabilities in presence of redundant packets and several traffic sources. Performance Evaluation **36-37** (1999) 486–518
15. Bolot, J.C., Crépin, H., García, A.V.: Analysis of audio packet loss in the Internet. NOSSDAV (1995)
16. Salamatian, M.R.: Transmission Multimédia Fiable Sur Internet. PhD thesis, Université Paris Sud UFR Scientifique d'Orsay (2000)
17. García, A.V.: Mécanismes de Contrôle pour la Transmission de l'Audio sur l'Internet. PhD thesis, Université de Nice Sophia-Antipolis (1996)
18. Figueiredo, D.R., de Souza e Silva, E.: Efficient mechanisms for recovering voice packets in the Internet. Globecom (1999)
19. Shenker, S.: Fundamental design issues for the future Internet. IEEE Journal on Selected Areas in Communications **13 (7)** (1995) 1176–1188
20. Kleinrock, L.: Queueing systems. John Wiley, New York (1976)
21. Cohen, J.W.: The Single Server Queue. North-Holland (1969)