Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier

LIRMM

Rapport de Recherche

No : 03-007

Avril 2003

Audio quality for a simple forward error correcting code

Y. Calas and A. Jean-Marie, calas,ajm@lirmm.fr

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier

161 rue Ada, 34392 Montpellier cedex 5

tel : (+33)4 67 41 85 85 – fax : (+33)4 67 41 85 00

Audio quality for a simple forward error correction code

Yvan Calas Alain Jean-Marie

Avril 2003

Abstract

The aim of this paper is to study the audio quality offered by a simple forward error correction (FEC) code used in audio applications like Freephone or Rat. This coding technique consists in adding to every audio packet a redundant information concerning a preceding audio packet which belongs to the same audio flow. We show that the audio quality depends not only on the number of FEC flows and the utility function associated to the quantity of information received, but also on the traffic conditions. Indeed, no improvement in the audio quality can be obtained for a smooth traffic whereas a marginal improvement can be observed for a bursty traffic. A significant increase of the audio quality is reached for a heavier bursty traffic. We also show that increasing the offset between the original audio packet and the packet bearing its redundancy does not improve significantly the audio quality.

Keywords: FEC, audio quality, utility function, Markov chain.

Résumé

Ce papier étudie la qualité audio offerte par un codage FEC simple utilisé dans les applications audio telles que Freephone ou Rat. Cette technique de codage consiste en l'ajout dans chaque paquet audio d'une information redondante concernant un paquet audio précédent appartenant au même flux audio. Nous montrons que la qualité audio dépend non seulement du nombre de flux FEC et de la fonction utilité associée à la quantité d'information reçue, mais également des conditions de trafic. En effet, aucune amélioration de la qualité audio est obtenue pour un trafic lisse alors qu'une faible amélioration est observée pour un trafic en rafale. Une amélioration significative de la qualité audio est obtenue pour un trafic en rafale avec une plus grande charge. Nous montrons également que le fait d'augmenter la distance entre le paquet audio original et le paquet portant sa redondance n'améliore pas de manière significative la qualité audio.

Mots-clés : FEC, qualité audio, fonction utilité, chaîne de Markov.

1 Introduction

Recent years have seen a growing use of audio and video applications in the Internet. Unlike file transfer applications like FTP or HTTP, these applications have strong real-time constraints. Yet, even though there is an increasing demand of a more predictable service, the current Internet offers only a best effort service without any performance guaranties on delay variations (jitter) and packet losses for instance.

The compensation for jitter can be accomplished through adaptive playout algorithms [15, 11, 17] and consists generally in taking some measurements on the delays experienced by packets. As for the packet losses, they can be handled through a variety of different forward error correction (FEC) algorithms and local repair at the receiver. Based on parity codes [18], Reed-Solomon codes [10, 16] or redundant speech codecs [8, 6, 5], these FEC algorithms send redundant information to compensate for loss.

In this paper, we focus on a simple FEC scheme (Fig. 1) which has been standardized by the IETF [13]. This scheme has already been used in audio tools like Freephone [20] and Rat [14], and has been generalized in [9]. It consists in adding a low quality copy of the original packet n to packet $n+\phi$ ($\phi \geq 1$). If packet n is lost in the network, it can be recovered and played out by the receiver if packet $n+\phi$ is correctly received. The redundant copy of packet n contained in packet $n+\phi$ is usually obtained by coding packet n with a lower-bandwidth rate, lower-quality encoding technique such as LPC or GSM. The quality of the reconstructed copy depends on the amount of information of packet n carried by packet $n+\phi$. The spacing between the original packet and its redundancy represented by the offset ϕ is a compromise between loss recovery and interactivity. Indeed, a large value of the parameter ϕ is expected to reduce the impact of correlated losses (as it is usually the case in the Internet [9, 3, 4]), but increases the delay of recovery and the jitter. A high delay would deteriorate the interactivity of a conversation and a high jitter would affect the fluidity of the speech.

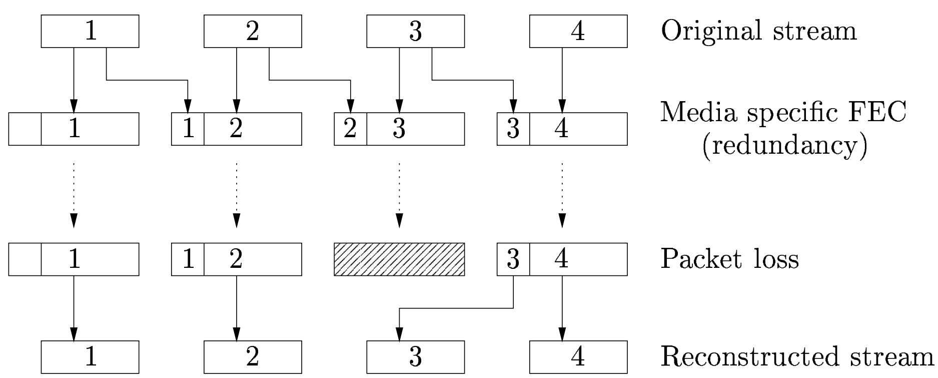

Figure 1: FEC scheme used in Freephone and Rat ($\phi = 1$)

There exist few analytical models for the computation of the performances of this FEC scheme : [2] has used a simple $M/M/1/K$ queuing system to study the audio quality for a single audio flow. This work has been extended in [1] by using a $M/G/1/K$ queuing system to study the audio quality for an audio flow which was multiplexed with an exogenous flow. It has been shown that the benefit of this simple FEC scheme depends on the number of FEC sources implementing it, and also on the shape of the utility function [19] which is not necessarily linear for multimedia applications.

In the present paper, we continue the work realized in [2, 1] and adapt it to the model which is described in [12] (and extended in [7]) in order to study the audio quality obtained from this FEC scheme. Under various conditions of traffic, we observe the behavior of the audio quality when a single audio flow is multiplexed with several non-FEC input flows generating an exogenous traffic. We also analyze the case where all flows become FEC audio flows. In addition, we study the effect of the offset ϕ on the audio quality.

The rest of the paper is organized as follows. In Section 2, we explain the analytical framework. Section 6 will focus on numerical results obtained from the model proposed in Section 3 using the metrics presented in Section 5. We propose in Section 4 two iterative algorithms which compute all the presented metrics and evaluate the complexity of each algorithm. Conclusions are drawn in Section 7.

2 Analytical framework

In this section, we recall the markovian model introduced in [12]. We briefly describe the overall topology of the system and the Markov chain.

2.1 Topology of the system

We consider a queue of capacity B packets in FIFO mode with N independent sources as input. One of the N sources (the FEC source) generates tagged packets (foreground traffic). The $N-1$ other sources (which may be FEC or non-FEC) generate a background traffic which interferes with the foreground traffic. At each slot, 0 up to N packets are generated by the N sources depending on their activities and are sent to the queue. Moreover at the beginning of each slot, one packet is served (if the queue is not empty). So the service time of every packet is one slot. We consider that the size of all the packets belonging to the audio flow or to the exogenous flow is fixed. We study in the following the tagged traffic generated by the FEC source.

2.2 Source model

The source model used in this paper is a discrete-time on/off model which alternates between active state (1) and idle state (0) periods. Let $q_{1,tt'}$ be the transition probability of one source from state t to state t' with $t, t' \in \{0, 1\}$, and let α and β be respectively the idle-to-idle and active-to-active state transition probabilities. At each slot, the state moves from the active to the idle state with probability $1-\beta$ and from the idle to the active state with probability $1-\alpha$. We assume that a state transition takes place just prior to the end of a time slot and that a packet is generated at the beginning of a slot if the new state is the active state. By construction, the length of a burst of packets is geometrically distributed and the probability for a burst to be of length k is equal to $\beta^{k-1}(1-\beta)$. Moreover, the average time for a source to stay in the active state (respectively in the idle state) is $1/(1-\beta)$ (respectively $1/(1-\alpha)$). Therefore, the stationary probability of the active state is the *normalized load* offered by one non-FEC source and is equal to $\rho_1 = (1-\alpha)/(2-\alpha-\beta)$.

Let ρ_{et} be the load offered by a FEC source and r be the ratio of redundancy volume and original packet volume (in general $r \in [0, 1]$). This load increases with the ratio r of redundancy generated by the FEC source :

$$\rho_{et} = \rho_1(1+r). \tag{1}$$

As r increases, the load offered by the FEC source becomes heavier than the load offered by one non-FEC source. This effect is obtained by increasing the active-to-active transition probability. This new transition probability is equal to :

$$\beta' = 2 - \alpha - (1 - \alpha)/\rho_{et}. \tag{2}$$

The *normalized aggregate load*, which is defined as the total load generated by N non-FEC sources is then computed as $\rho = N\rho_1$, since the sources are identical in the absence of coding. When the traffic generated by one FEC source is mixed with $N-1$ non-FEC sources, the *aggregate load* becomes equal to $\rho = \rho_{et} + (N-1)\rho_1$. Moreover, if the N sources generate audio packets, each source increases its normalized load by a factor of $1 + r$ (as seen in (1)) and the aggregate load will be equal to $\rho = N\rho_1(1 + r)$.

2.3 The Markov chain

The Markov chain described by Oguz and Ayanoglu in [12] evolves in a state space $\mathcal{S}$ where each state is represented by a triple (b, t, u). For a given slot, $b \in \{0, \ldots, B\}$ is the number of packets contained in the queue having a buffer capacity of size B, $t \in \{0, 1\}$ is the state (0 if idle, 1 if active) of the tagged (FEC) source in the slot and $u \in \{0, \ldots, N-1\}$ is the number of non-tagged sources which generate a packet in the slot. Some states are considered inconsistent due to the boundary conditions at $b = 0$ or $b = B$. Therefore, $\mathcal{S}$ is defined as follows :

$$\mathcal{S} = \big\{(b, t, u) : t + u \leq b + 1 \text{ if } b < B, \text{ and } t + u > 0 \text{ if } b = B\big\}.$$

$\mathcal{S}$ can be partionned into two subsets : states where the FEC source is idle (therefore there is no loss of tagged packets) and states where the FEC source is active (tagged packet may then be lost) :

$$\mathcal{S}_D = \big\{s = (b, 0, u) \text{ with } s \in \mathcal{S}\big\}.$$

Among the states in $\mathcal{S} - \mathcal{S}_D$ (that is when the FEC source is active), we have two cases according to the loss or saving of the tagged packet :

$$\mathcal{S}_S = \big\{s = (b, 1, u) \text{ and the tagged packet is saved}\big\},$$

$$\mathcal{S}_L = \big\{s = (B, 1, u) \text{ and the tagged packet is lost}\big\}.$$

$\mathcal{S}_L$ adds new states in the Markov chain. We define by $\mathcal{S}_E = \mathcal{S}_D \cup \mathcal{S}_S \cup \mathcal{S}_L$ the set of all the states of the Markov chain. Moreover, the transition probability q_{ij} from state $i = (b, t, u)$ to state $j = (b', t', u')$ is readily computed from α and β. Details can be found in [12, 7].

3 Loss rate modeling

We propose in this section a novel recursive formula to compute the packet loss rate, using the Markov chain introduced in Section 2. For this purpose, we define by $g_i^{(\phi)}(m, 1; l, 1)$ the probability that packet n is lost (if $m = 1$) or saved (if $m = 0$) and that packet $n + \phi$ is lost (if $l = 1$) or saved (if $l = 0$). Conditioning on the first transition of the Markov chain, we obtain the recursive formula (3) where $\mathbb{1}_{\{A\}}$ is the event-indicator function which is equal to 1 if condition A is true and is equal to 0 otherwise. For all $i \in \mathcal{S}_E$,

$$g_i^{(\sigma)}(m,k;l,h) = \sum_{j \in \mathcal{S}_D} q_{ij} g_j^{(\sigma)}(m,k;l,h) + b_i, \tag{3}$$

where :

$$\begin{aligned} b_i &= \sum_{j \in \mathcal{S}_S} q_{ij} g_j^{(\sigma-1)}(0,0;l,h)\mathbb{1}_{\{\sigma=\phi,m=0,k=1\}} \\ &+ \sum_{j \in \mathcal{S}_L} q_{ij} g_j^{(\sigma-1)}(0,0;l,h)\mathbb{1}_{\{\sigma=\phi,m=k=1\}} \\ &+ \sum_{j \in \mathcal{S}_S} q_{ij} g_j^{(\sigma)}(0,0;0,0)\mathbb{1}_{\{\sigma=0,l=0,h=1\}} \\ &+ \sum_{j \in \mathcal{S}_L} q_{ij} g_j^{(\sigma)}(0,0;0,0)\mathbb{1}_{\{\sigma=0,l=h=1\}} \\ &+ \sum_{j \in \mathcal{S}_S \cup \mathcal{S}_L} q_{ij} g_j^{(\sigma-1)}(0,0;l,h)\mathbb{1}_{\{0<\sigma<\phi\}}. \end{aligned} \tag{4}$$

Equation (3) makes a clear distinction between the original audio packet n and the packet $n+\phi$ bearing a compressed encoded audio information of packet n.

The loss of an audio packet generated by the tagged source depends on the type of the arrival state j reached at the next slot in the Markov chain. If the tagged source is inactive, then there is no generation of an audio packet. The value of $g_j^{(\sigma)}(m,k;l,h)$ should therefore be computed. On the other hand, if the tagged source is active then a tagged packet is generated. This generated tagged packet could be :

1. packet n (i.e. $\sigma = \phi$) :
   (a) In the case where the tagged packet is saved ($j \in \mathcal{S}_S$), we compute $g_j^{(\sigma-1)}(m,0;l,1)$ provided that $m=0$ and $k=1$.
   (b) In the case where the tagged packet is lost ($j \in \mathcal{S}_L$), we compute $g_j^{(\sigma-1)}(m-1,0;l,1)$ provided that $m=k=1$.
2. packet $n+\phi$ (i.e. $\sigma = 0$) :
   (a) In the case where the tagged packet is saved ($j \in \mathcal{S}_S$), we compute $g_j^{(\sigma)}(0,0;l,0)$ provided that $l=0$ and $h=1$.
   (b) In the case where the tagged packet is lost ($j \in \mathcal{S}_L$), we compute $g_j^{(\sigma)}(0,0;l-1,0)$ provided that $l=h=1$.
3. a packet different than packet n and packet $n+\phi$ (i.e. $0<\sigma<\phi$). In this case, we compute $g_j^{(\sigma-1)}(0,0;l,1)$ without any additional condition.

For all $i \in \mathcal{S}_E$ and $m,l \in \{0,1\}$, once the values of $g_i^{(\phi)}(m,1;l,1)$ are computed, the probability to lose or save packet n and also to lose or save packet $n+\phi$ is given by :

$$G^{(\phi)}(m;l) = \frac{1}{\rho_{et}} \sum_{i \in \mathcal{S}_S \cup \mathcal{S}_L} p_i g_i^{(\phi)}(m,1;l,1), \tag{5}$$

where p_i is the stationary probability to be in state i of the Markov chain.

4 Algorithmic issues

In this section, we present an algorithm to compute $g_i^{(\phi)}(m,1;l,1)$ according to (3) and another algoritm to compute the audio quality using (9) by varying the redundancy parameter r. We finally conclude this section by studying the complexity of these two algorithms.

4.1 Algorithm for computing $g_i^{(\phi)}(m,1;l,1)$

Algorithm 1 is an iterative algorithm which computes all the $g_i^{(\phi)}(m,1;l,1)$ for the FEC scheme presented in Section 1. The algorithm is composed of four steps. In the first step, we initialize $g_i^{(\sigma=0)}(0,0;0,0)$ to 1. We then compute the LU factorization of matrix A. The value of A and the detailed explanation of this factorization will be given in Section 4.3 which deals with the complexity of the algorithm. In the second step, assuming we are in the state i in the Markov chain, we compute $g_i^{(\sigma=0)}(0,0;l,1)$, the probability of losing or saving packet $n+\phi$ which bear the redundancy for the packet n. Note that for this computation, packet n is not taken into account. The third step consists in computing $g_i^{(\sigma=\phi)}(0,0;l,1)$ since we have to take into account the offset ϕ between packet n and packet $n+\phi$. Finally, the purpose of step four is to compute $g_i^{(\sigma=\phi)}(m,1;l,1)$ in order to obtain the probability to lose or to save packet n and packet $n+\phi$ given a state i of the Markov chain.

4.2 Algorithm for computing audio quality as a function of redundancy r

Algorithm 2 consists in computing the audio quality given an offset ϕ between the original packet and the packet bearing a compressed encoded audio information of the original packet. The audio quality $Q(\phi,r)$ is deduced by computing the probability distribution $G^{(\phi)}(m;l)$ for various values of the redundancy parameter r and then by using (9).

4.3 Complexity of algorithms

Let now consider the complexity of Algorithms 1 and 2. The computation of $g_i^{(\sigma)}(m,k;l,h)$ for $i \in \mathcal{S}_D$ in steps 2, 3 and 4 of Algorithm 1 requires the resolution of a set of $|\mathcal{S}_D|$ linear equations with $|\mathcal{S}_D|$ unknowns. These equations can be written in a matrix form $Ax=b$ where $A=(A_{ij})_{(i,j)\in\mathcal{S}_D\times\mathcal{S}_D}$ is a sparse matrix having the following coefficients :

$$A_{ij} = \begin{cases} 1-q_{ii} & \text{if } i=j, \\ -q_{ij} & \text{if } i\neq j. \end{cases} \tag{6}$$

The vector $b=(b_i)_{i\in\mathcal{S}_D}$ is defined by (4) and the vector $x=(g_i^{(\sigma)}(m,k;l,h))_{i\in\mathcal{S}_D}$ is the solution of the set of linear equations.

In order to reduce the complexity of the resolution of the system of linear equations, we compute first the LU factorization of matrix A (using Gaussian elimination) with complexity of at most $O(|\mathcal{S}_D|^3)$. Once the LU factorization of A is known, the resolution of the system of linear equations can be performed with complexity of $O(|\mathcal{S}_D|^2)$ instead of $O(|\mathcal{S}_D|^3)$. Moreover, $g_i^{(\sigma=0)}(0,0;l,1)$, $g_i^{(\sigma)}(0,0;l,1)$ for $1\leq\sigma\leq\phi-1$ and $g_i^{(\sigma=\phi)}(m,1;l,1)$, for $i\in\mathcal{S}_S\cup\mathcal{S}_L$, can be computed directly from (3) with an insignificant complexity.

Algorithm 1 Computation of $g_i^{(\phi)}(m,1;l,1)$

```
/* First step : initialization */
∀i ∈ S_E, g_i^(σ=0)(0,0;0,0) = 1.
Compute the LU factorization (Gaussian elimination) of the matrix A.
/* Second step : computation of g_i^(σ=0)(0,0;l,1), ∀i ∈ S_E, ∀l ∈ {0,1} */
for l = 0 to l = 1 do
  Solve g_i^(σ=0)(0,0;l,1) for all i ∈ S_D using (3) and the LU factorization of A.
end for
for l = 0 to l = 1 do
  Compute g_i^(σ=0)(0,0;l,1) for all i ∈ S_S ∪ S_L using (3).
end for
/* Third step : computation of g_i^(σ)(0,0;l,1), ∀i ∈ S_E, ∀l ∈ {0,1}, ∀σ ∈ {1,...,φ−1}
*/
if φ > 1 then
  for σ = 1 to σ = φ − 1 do
    for l = 0 to l = 1 do
      Solve g_i^(σ)(0,0;l,1) for all i ∈ S_D using (3) and LU factorization of A.
    end for
    for l = 0 to l = 1 do
      Compute g_i^(σ)(0,0;l,1) for all i ∈ S_S ∪ S_L using (3).
    end for
  end for
end if
/* Fourth step : computation of g_i^(σ=φ)(m,1;l,1), ∀i ∈ S_E, ∀m,l ∈ {0,1} */
for l = 0 to l = 1 do
  for m = 0 to m = 1 do
    Solve g_i^(σ=φ)(m,1;l,1) for all i ∈ S_D using (3) and LU factorization of A.
  end for
  for m = 0 to m = 1 do
    Compute g_i^(σ=φ)(m,1;l,1) for all i ∈ S_S ∪ S_L using (3).
  end for
end for
```

Algorithm 2 Computation of audio quality as function of the redundancy parameter r.

```
Require: A variable step with step ∈]0,1[.
  r = 0.
  while r ≤ 1 do
    Compute g_i^(φ)(m,1;l,1) for all i ∈ S_E using Algorithm 1.
    Compute G^(φ)(m;l) using (5) for m,l ∈ {0,1}.
    Compute PLRBC(φ,r) using (7).
    Compute audio quality Q(φ,r) using (9).
    r = r + step.
  end while
```

Consequently, the complexity of Algorithm 1 is $O\big(|\mathcal{S}_D|^3 + 4|\mathcal{S}_D|^2 + 2(\phi - 1)|\mathcal{S}_D|^2\big)$, which can be rewritten as $O\big(|\mathcal{S}_D|^3 + 2|\mathcal{S}_D|^2(1+\phi)\big)$. Based on this complexity we then deduce the complexity of Algorithm 2 which is $O\big(\frac{1}{step}(|\mathcal{S}_D|^3 + 2|\mathcal{S}_D|^2(1+\phi))\big)$.

This complexity can be improved by taking into account the following remark : in case of only one FEC source and $N-1$ non-FEC sources, the matrix A remains unchanged for every value of r (and therefore every value of β', see (2)), i.e. $\forall r \in \mathbb{N}, A(r) = A$. The LU factorization of A is only computed once before executing Algorithm 2 and is not recomputed in Algorithm 1 for each value of r. Consequently, the complexity of Algorithm 2 is reduced to $O\big(|\mathcal{S}_D|^3 + \frac{1}{step}2|\mathcal{S}_D|^2(1+\phi)\big)$. However, this remark is not valid if all sources are FEC sources because the matrix A is modified according to the value of r : $\forall r, r' \in \mathbb{N}$ with $r \neq r', A(r) \neq A(r')$. As a result, the LU factorization of A has to be computed for every value of r, if all sources are FEC sources.

5 Metrics

Let Z_n be a random variable that represents a loss or a successful arrival to the destination of packet n issued from the audio flow. We define Z_n as :

$$Z_n = \begin{cases} 0 & \text{if packet } n \text{ is not lost,} \\ 1 & \text{if packet } n \text{ is lost.} \end{cases}$$

5.1 Packet loss rate

We define the packet loss rate (PLR) metric. When the Markov chain is stationary, each packet has the same loss probability :

$$PLR(r) = EZ_n = \sum_{l=0}^{1} G^{(\phi)}(1;l). \tag{7}$$

5.2 Audio quality

We denote by $Q(\phi, r)$ the average audio quality obtained after the reconstruction of the lost original packet from the audio packet bearing the redundant information concerning the lost packet. Since audio quality does not necessarily increase linearly with the volume of data in a packet [19], Altman *et al.* [1] have introduced an utility function U which depends on the quantity of information r, and is such that $U(0) = 0$ and $U(1) = 1$. As shown in [1], we have :

$$Q(\phi, r) = P(Z_n = 0) + U(r)P(Z_n = 1)P(Z_{n+\phi} = 0|Z_n = 1). \tag{8}$$

Assuming that the Markov chain is in the steady state, the probability to lose packet n coincides with the stationary probability to lose any packet, in other words, with the PLR computed by (7). Next, the probability $P(Z_{n+\phi} = 0|Z_n = 1)$ is the probability to lose packet n and also to save packet $n + \phi$. Consequently, we have :

$$P(Z_{n+\phi} = 0|Z_n = 1) = G^{(\phi)}(1;0).$$

Finally, substituting these values in (8) leads to :

$$Q(\phi, r) = 1 - PLR(r)(1 - U(r)G^{(\phi)}(1;0)). \tag{9}$$

Observe that for all $\phi \geq 1$, $Q(\phi, 0) = Q(0) = 1 - PLR(0)$. In the experiments below, we shall use two utility functions U_0 and U_m taken or adapted from [1] which represent extreme cases :

- $U_0(r) = r$,
- $U_m(r) = \begin{cases} 0 & \text{if } r = 0, \\ 1 & \text{if } r > 0. \end{cases}$

Other utility functions used in the experiments of [1] lie between these two ones, and are omitted to simplify the presentation of our results. The corresponding quality functions lie between those we obtain. The function U_0 represents the case of an utility proportional to the quantity of information. The function U_m represents an ideally optimistic case where only a small amount of information provides the maximum level of utility, and gives the best upper bound on the maximum audio quality that can be obtained. Since $\lim_{r \to 0^+} U_m(r) = 1$ and since in this case quality appears to be decreasing with r (see next section), then it is possible to quantify the maximum audio quality Q^* that can be obtained by :

$$Q^*(\phi) = 1 - PLR(0)(1 - G^{(\phi)}(1; 0)),$$

and the maximum improvement by :

$$Q^*(\phi) - Q(0) = PLR(0)G^{(\phi)}(1; 0).$$

We show in Section 6 that for certain conditions of traffic we can obtain significant improvement of the audio quality by increasing slightly the redundancy r.

6 Numerical Results

In this section, we present numerical results showing the variation of the audio quality according to the redundancy parameter r and the offset ϕ. These results are obtained for a buffer size $B = 25$, for $N = 16$ sources and for different types of traffic depending on the configuration of the coefficients α and β. We set up $\alpha = 0.995$ and $\beta = 0.905$ in order to obtain a *bursty traffic* (subsection 6.1) which gives for each source a normalized load of $\rho_1 = 0.05$. A *heavier bursty traffic* is generated for $\alpha = 0.99$ and $\beta = 0.91$ (subsection 6.2) where $\rho_1 = 0.1$. Finally, we set up $\alpha = 0.96$ and $\beta = 0.24$ (and therefore $\rho_1 = 0.05$) so as to obtain a *smooth traffic* (subsection 6.3) with bursts of average length 1.3 separated by idle periods of average length 25.

6.1 Bursty traffic

The bursty traffic generated by one FEC source and $N - 1$ non-FEC sources (Fig. 2) gives an aggregate load $\rho \in [0.8, 0.85]$. We can observe that the curves of the utility functions U_0 and U_m are close to each other. However, the audio quality for U_0 decreases monotonously with r, whereas the function U_m improves the audio quality for small amounts of redundancy ($r < 0.2$) and decreases monotonously with r. This means that the addition of a small amount of redundancy in every audio packet (specifically using a LPC codec) can slightly increase the audio quality provided that the utility is close to U_m. This improvement peaks at 90.2% for $\phi = 1$. Furthermore, we can observe that the increase of the offset

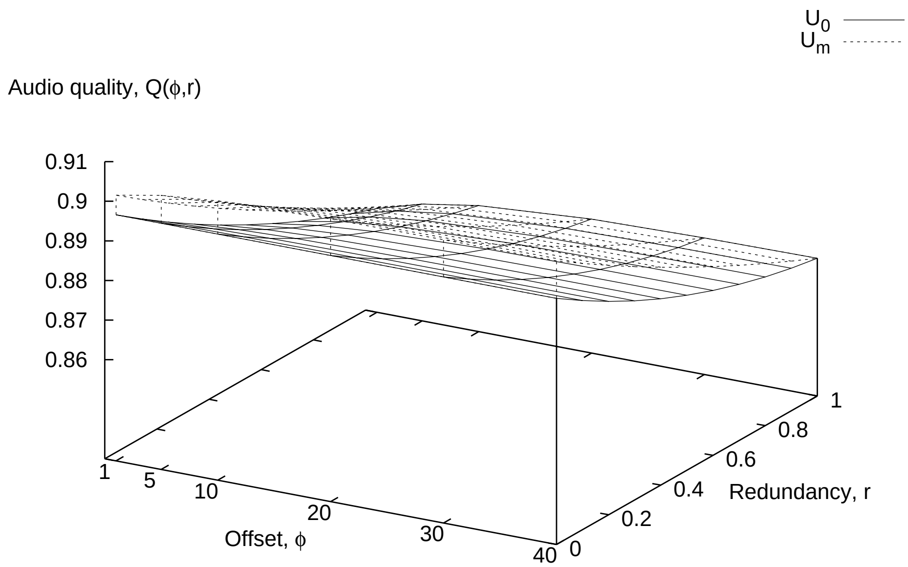

Figure 2: Audio quality for a bursty traffic with one FEC source.

ϕ is not effective for both utility functions : $\max_r Q(40,r) - Q(1,r) = 0.008$ for U_0 and $Q^*(40) - Q^*(1) = 0.004$ for U_m.

For the configuration of Fig. 3 where all sources are FEC sources, we obtain a heavier aggregate load since $\rho \in [0.8, 1.6]$. In this case the curves for U_0 and U_m are almost identical and therefore the dependence on the utility function becomes less important than in the case of the configuration of Fig. 2. Moreover, the audio quality decreases with r more quickly than the curves in Fig. 2 (for instance, for $\phi = 1$, it decreases from 0.897 for $r = 0$ to 0.648 for $r = 1$), and the increase of ϕ does not improve significantly the audio quality.

6.2 Heavier bursty traffic

The heavier bursty traffic generated by one FEC source and $N-1$ non-FEC sources (Fig. 4) overloads the system with an aggregate load $\rho \in [1.6, 1.7]$. In this case, of course, we obtain a lower audio quality as compared to audio quality obtained for a bursty traffic. However, it is possible to increase the audio quality and to obtain an important improvement as compared to the one observed in the case of a bursty traffic. For the function U_0, the audio quality increases uniformly with r. This means that the audio quality can be improved by the use of a more powerful audio coder that can generate the redundancy contained in each audio packet. On the other hand, the function U_m increases significantly the audio quality when a small amount of redundancy is added in every audio packet by the use of LPC or GSM codecs. Nevertheless, the addition of more redundancy decreases the audio quality. Moreover, it is more beneficial to increase the offset ϕ in this case than in the case of a bursty traffic : $\max_r Q(40,r) - Q(1,r) = 0.034$ for U_0 and $Q^*(40) - Q^*(1) = 0.033$ for U_m. This suggests increasing the offset ϕ as much as the application allows. On the other

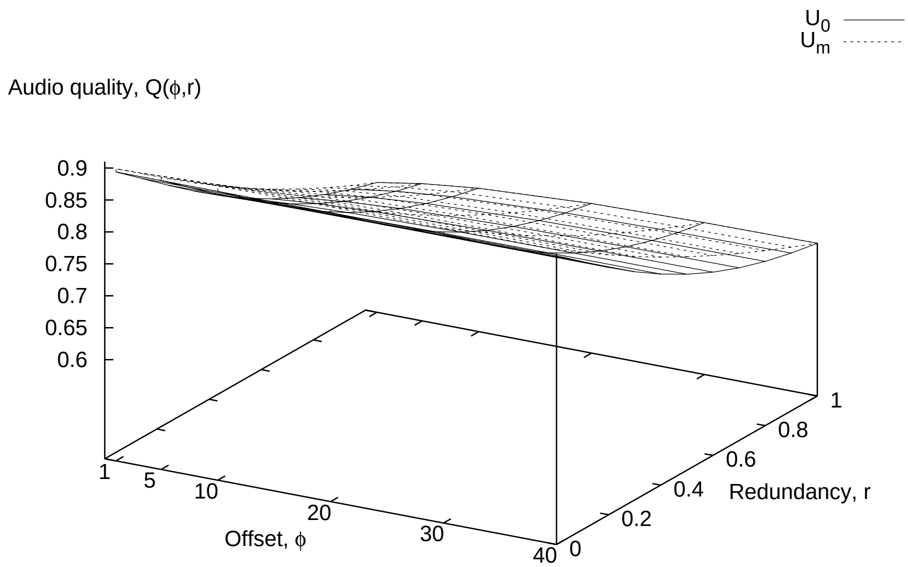

Figure 3: Audio quality for a bursty traffic with N FEC sources.

hand, the improvement with respect to ϕ quickly reaches a plateau. Increasing ϕ above 10 is not useful anymore.

When all sources become FEC sources (Fig. 5), the aggregate load is $\rho \in [1.6, 3.2]$. The addition of redundancy allows a significant increase of the audio quality for the function U_m. This is not the case for the function U_0 since it decreases continuously with r. Finally, it should be noted that in spite of a heavy network load, it is possible to improve the audio quality.

6.3 Smooth traffic

For the configuration of Fig. 6 when N FEC sources generate a smooth traffic with an aggregate load of $\rho \in [0.8, 1.6]$, we observe that whatever the utility function used, the addition of redundancy in each audio packet decreases monotonously the audio quality with r. Without the use of FEC, the loss rate is very small and the quality is close to one. The audio quality does not really need improvement in this case, and FEC deteriorates the quality of service. Similar conclusions can be drawn for the case of one FEC source and $N-1$ non-FEC sources (as shown in Fig. 6) when the aggregate load is $\rho \in [0.8, 0.85]$.

7 Conclusion

In this paper, we studied the performance of a simple FEC scheme implemented in recent audio tools like Rat [14] and Freephone [20]. This scheme consists in adding a low quality copy of the original audio packet n to the audio packet $n + \phi$. Our model is based on the Markov chain presented in [12] and on a recursive formula which computes the probability to lose or to save packet n and packet $n + \phi$.

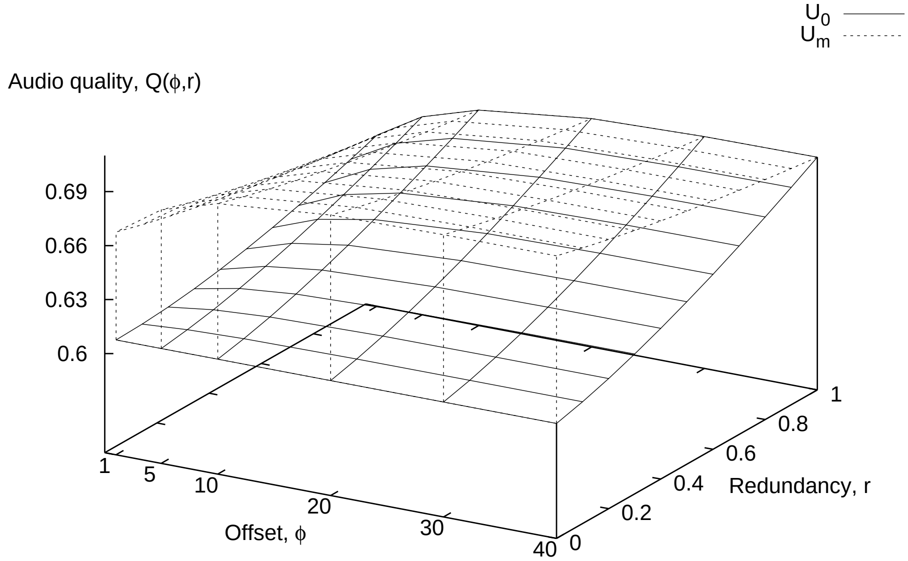

Figure 4: Audio quality for a heavier bursty traffic with one FEC source.

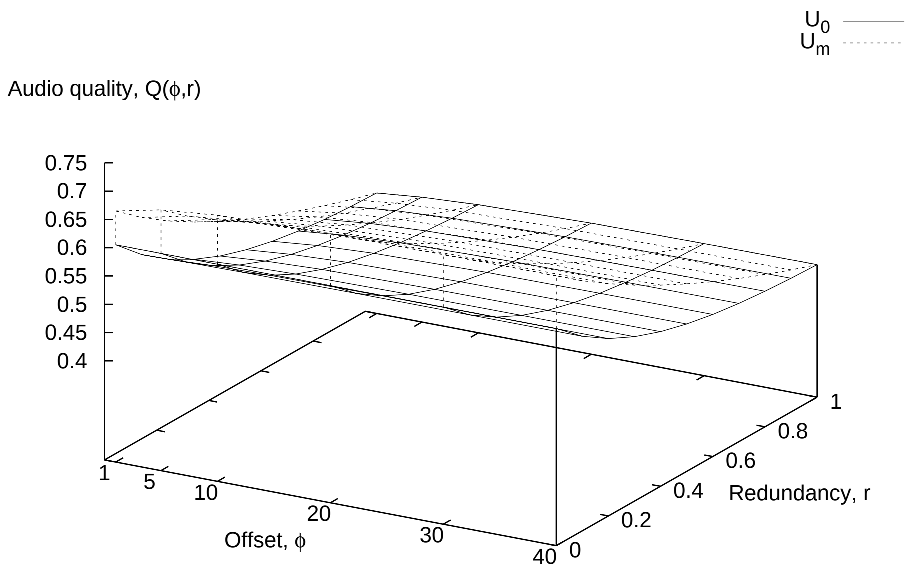

Figure 5: Audio quality for a heavier bursty traffic with N FEC sources.

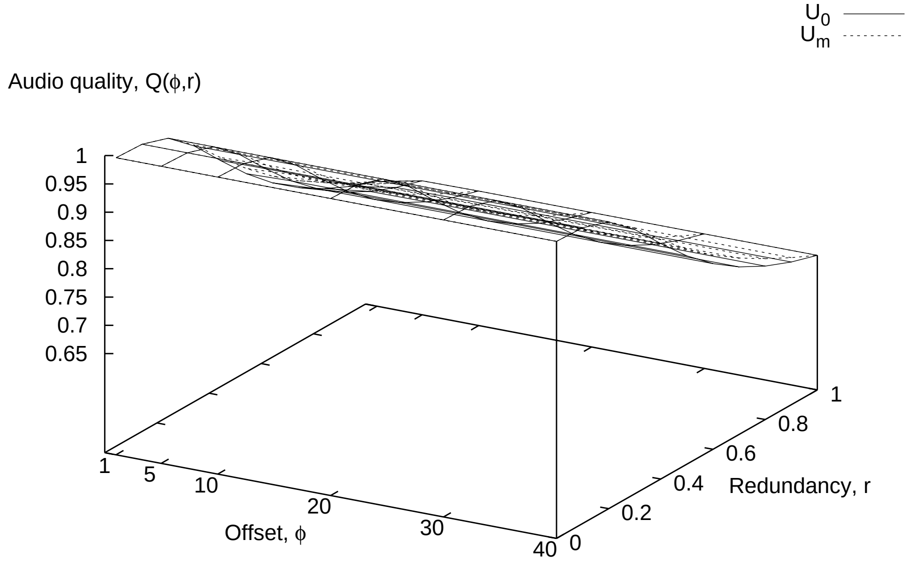

Figure 6: Audio quality for a smooth traffic with N FEC sources.

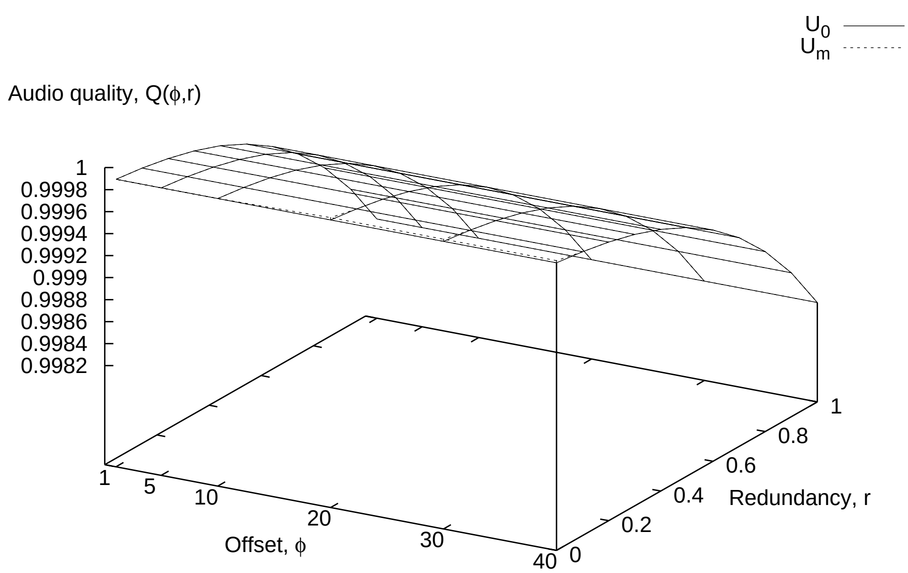

Figure 7: Audio quality for a smooth traffic with one FEC source.

We showed that the FEC scheme can improve the audio quality depending not only on the number of FEC flows and the utility function as shown in [2, 1] but also on the traffic conditions. The results showed that a significant improvement of the audio quality is only observed in the case of a heavier bursty traffic. This traffic condition illustrated that unlike [1], the use of a linear utility function can increase audio quality. Even though the increase of the audio quality for the case where all sources are FEC depends on the utility function as shown in [1], it is only for a heavier bursty traffic that a significant improvement is obtained. Furthermore as in [1], we observed that quality increases with ϕ. However, it is interesting to increase ϕ for a heavier bursty traffic rather for a bursty or a smooth traffic. Yet, an increase above a certain threshold gives a marginal improvement. In addition, we showed that no improvement in audio quality can be obtained for a smooth traffic. This confirms results of [1]. However, the quality observed in the absence of redundancy is close to one in that case. Finally in the case of a bursty traffic, a light or marginal improvement is obtained depending on the number of FEC flows. Nevertheless the quality decreases with r and FEC should be used with as few redundancy as possible, depending on the actual shape of the utility function.

References

[1] E. Altman, C. Barakat, and V.M. Ramos. On the utility of FEC mechanisms for audio applications. In *Proc. of QofIS'2001, Second International Workshop on Quality of Future Internet Services*, Coimbra, Portugal, September 2001.

[2] E. Altman, C. Barakat, and V.M. Ramos. Queueing analysis of simple FEC schemes for IP telephony. In *Proc. IEEE INFOCOM'01*, Anchorage, Alaska, April 2001.

[3] E. Altman and A. Jean-Marie. Loss probabilities for messages with redundant packets feeding a finite buffer. *IEEE Journal on Selected Areas In Communications*, 16(5):778–787, June 1998.

[4] J.-C. Bolot, H. Crepin, and A. Vega-Garcia. Analysis of audio packet loss in the Internet. In *NOSSDAV*, 1995.

[5] J.-C. Bolot and A. Vega-Garcia. The case for FEC-based error control for packet audio in the Internet. ftp://ftp-sop.inria.fr/rodeo/bolot/96.FEC_audio.ps.Z, 1996.

[6] J.-C. Bolot and A. Vega-Garcia. Control mechanisms for packet audio in the Internet. In *Proc. IEEE INFOCOM'96*, pages 232–239, San Fransisco, CA, USA, April 1996.

[7] Y. Calas and A. Jean-Marie. On the efficiency of forward error correction at the packet level. Technical report, LIRMM, University of Montpellier II, March 2002.

[8] V. Hardman, M. A. Sasse, M. Handley, and A. Watson. Reliable audio for use over the Internet. In *International Networking Conference (INET)*, September 1995.

[9] I. Kouvelas, O. Hodson, V. Hardman, and J. Crowcroft. Redundancy control in real-time Internet audio conferencing. In *Proc. of AVSPN*, 1997.

[10] A.J. McAuley. Reliable broadband communications using a burst erasure correcting code. In *Proc. ACM SIGCOMM'90*, pages 297–306, Philadelphia, PA, USA, September 1990.

[11] S.B. Moon, J. Kurose, and D. Towsley. Packet Audio Playout Delay Adjustment : Performance Bounds and Algorithms. *ACM/Springer Multimedia Systems*, 5(1):17–28, January 1998.

[12] N.C. Oguz and E. Ayanoglu. Performance analysis of two-level forward error correction for lost cell recovery in ATM networks. In *INFOCOM'95 (2)*, pages 728–737, 1995.

[13] C. Perkins, I. Kouvelas, O. Hodson, and V. Hardman. RTP payload for redundant audio data. RFC 2198, January 1997.

[14] The Mice Project. Rat : Robust audio tool. http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/, Multimedia Integrated Conferencing for European Researchers, University College London.

[15] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne. Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks. In *Proc. IEEE INFOCOM'94*, pages 680–688, Torento, Canada, June 1994.

[16] L. Rizzo. Effective Erasure Codes for Reliable Computer Communication Protocols. *Computer Communication Review*, 27(2):24–36, April 1997.

[17] J. Rosenberg, L. Qiu, and H. Schulzrinne. Integrating packet FEC into adaptive voice playout buffer algorithms on the Internet. In *Proc. IEEE INFOCOM'00*, Tel Aviv, Israel, March 2000.

[18] N. Shacham and P. McKenney. Packet recovery in high-speed networks using coding and buffer management. In *Proc. INFOCOM 90*, volume 1, pages 124–131, San Francisco, CA, USA, June 1990.

[19] S. Shenker. Fundamental design issues for the future Internet. *IEEE Journal on Selected Areas in Communications*, 13(7):1176–1188, 1995.

[20] A. Vega-Garcia and S. Fosse-Parisis. Freephone audio tool. http://www-sop.inria.fr/rodeo/ fphone/, High-Speed Networking Group, INRIA Sophia Antipolis.