Queuing analysis of simple FEC schemes for Voice over IP ⋆

Eitan Altman [a,1], Chadi Barakat [a], Víctor M. Ramos R. [b,2]

[a] *INRIA Sophia Antipolis, 2004 Route des Lucioles BP 93*
06902 Sophia Antipolis, France

[b] *Institut Eurécom, 2229 Route des Crêtes*
06904 Sophia Antipolis, France

Abstract

In interactive voice applications, FEC allows to recover from losses. FEC schemes need to be simple in order to allow interactivity. We study a simple scheme implemented in [9,17], where for every packet n, redundant information is added in packet $n+\phi$. The quality of a reconstructed copy depends on the amount of redundancy. Using queuing analysis we obtain simple expressions for the audio quality as a function of the amount of redundancy. We show that the expected amount of useful information deteriorates for any amount of FEC and any ϕ. We find conditions under which FEC leads to quality improvement.

Key words: Audio quality, voice over IP, ballot theorem, utility function, queuing analysis.

⋆ This paper is largely based on research results reported in the proceedings of IEEE Infocom 2001 (where we analyzed the usefulness of FEC in terms of useful information) and reported in the proceedings of the Second International Workshop on Quality of future Internet services (QofIS 2001) (where we studied the impact of FEC on the utility functions of voice over IP).

[1] E. Altman and C. Barakat are with INRIA, Sophia Antipolis, France. Email: {altman,cbarakat}@sophia.inria.fr. E. Altman is also with C.E.S.I.M.O., Universidad de Los Andes, Facultad de Ingeniería, Mérida, Venezuela. Email:altman@ing.ula.ve.

[2] V. Ramos is a Ph.D. student at Institut Eurecom. This work was done while he was realizing his DEA internship at INRIA. The author is also an associate professor at Universidad Autónoma Metropolitana in Mexico City. Email:Victor.Ramos@eurecom.fr.

1 Introduction

Real–time audio transmission is now widely used over the Internet and has become a very important application. Audio quality is still however an open problem due to the loss of audio packets and the variation of end–to–end delay (jitter). These two factors are a natural result of the simple best effort service provided by the current Internet. Indeed, the Internet provides a simple packet delivery service without any guarantee on bandwidth, delay or drop probability. The audio quality deteriorates (noise, poor interactivity) when packets cross a loaded part of the Internet. In the wait for some QoS facilities from the network like resource reservation, call admission control, etc., the problem of audio quality must be studied and solved on an end–to–end basis. Some mechanisms must be introduced at the sender and/or at the receiver to compensate for packet losses and jitter. The jitter is often solved by some adaptive playout algorithms at the receiver. Adaptive playout mechanisms are treated in detail in [18], and more recently in [19]. In this paper we focus on the problem of recovery from audio packet losses.

Mechanisms for recovering from packet losses can be classified as open loop mechanisms, or closed loop mechanisms [14]. Closed loop end–to–end mechanisms like ARQ (Automatic Repeat reQuest) are not adequate for real–time interactive applications since they increase considerably the end–to–end delay due to packet retransmission [3] . Open loop mechanisms like FEC (Forward Error Correction) are better adapted to real–time applications given that packet losses are recovered without the need of a retransmission. Some redundant information is transmitted with the basic audio flow. Once a packet is lost, the receiver uses (if possible) the redundant information to reconstruct the lost information. FEC schemes are recommended whenever the end–to–end delay is large so that a retransmission deteriorates the end–to–end quality.

An audio conversation is considered to be *interactive* if the two–way end–to–end delay is less than 250 ms, including media coding and decoding, network transit and playout buffering [13]. Moreover, an end–to–end delay less than 250 ms does not have any negative impact on the quality. We shall assume that this limit on end–to–end delay is respected, and that the loss of packets is the only source of deterioration of audio quality.

FEC has been often used for loss recovery in audio communication tools. It is a sender–based repair mechanism. An efficient FEC scheme is one that is able to repair most of packet losses. Now, when FEC fails to recover from a loss, applications can resort to other receiver–based repair mechanisms like insertion, interpolation, or regeneration, using well known methods [14].

[3] Note however, that ARQ could perform well as part of a link level protocol for some short links

The FEC schemes proposed in the literature are often simple, so that the coding and the decoding of the redundancy can be quickly done without impacting the interactivity. In particular, the redundancy is computed over small blocks of audio packets. Well known audio tools as Rat [17], and Freephone [9], generally work by adding some redundant information of (i.e., a copy of) packet n to the next packet $n+1$, so that if packet n is dropped in the network, it could be recovered and played out in case packet $n+1$ is correctly received. The redundant information carried by a packet is generally obtained by coding the previous packet with a code of lower rate than that of the code used for coding the basic audio flow. For example, a basic audio packet can be coded with PCM and its copy with GSM [21] or LPC [24]. Thus, if the reconstruction succeeds, the lost packet is played out with a copy coded at a lower rate. This has been shown to give better quality than playing nothing at the receiver. Figure 1 depicts this simple FEC scheme.

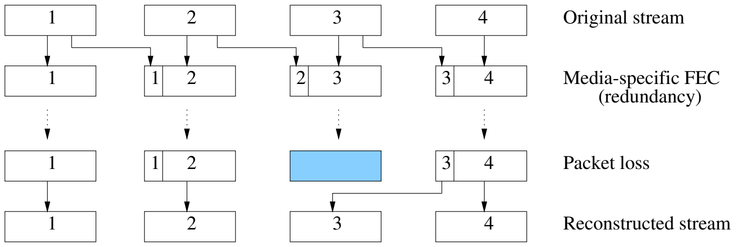


Fig. 1. The simple FEC mechanism where packet $n+1$ carries redundant information of packet n.

In this paper we address the problem of audio quality under this FEC scheme. In all the paper, when we talk about FEC in general, it is this particular scheme that we mean. We evaluate analytically the audio quality at the destination as a function of the parameters of the FEC scheme, of the basic audio flow and of the network. The performance of this FEC scheme has been evaluated via simulations [16,15], and tools like Freephone and Rat have implemented it. In [12], the authors propose to increase the offset between the original packet and its redundancy. They claim that the loss process in the Internet is bursty and thus, increasing the offset could give better performance than having the redundancy placed in the packet following immediately the original one. However, the authors in [12] did not propose any analytical expression that permits to study the impact of this spacing on the audio quality.

The paper is organized in two parts. In the first part, we use probabilistic methods and a Ballot theorem [23] to find an explicit expression for the audio quality in the case of a general offset not necessarily equal to one. The audio quality is supposed to be proportional to the volume of data received. We consider a single bottleneck node for the network and we focus 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). These assumptions hold when all flows in the network implement FEC, or when a round–robin scheduler with per–flow queuing is used. Under these assumptions, our analysis shows 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 scheme leads always to a deterioration of quality caused by an important increase in network load. Similar negative results have been already obtained using analytical tools for more sophisticated FEC schemes, see [7,1,10]. We consider in this paper both the case in which adding redundant information does not change the amount of useful information in the packet, and thus the total size of the packet increases with redundancy (along with appropriate scaling of loss probabilities and buffering capabilities), as well as the case in which the total size of the packet does not change, so that adding redundancy results in decreasing the transmitted rate of useful information.

In the second part of the paper, 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. We consider two aspects that may contribute to quality improvement: multiplexing with other flows and using quality functions which are not proportional to the volume of well received data (goodput). The expected quality is computed by using a *utility function* that indicates the audio quality at the receiver as a function of the transmission rate. In the previously used linear utility function we 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 depends on a large number of parameters. Yet some simplified non–linear utility functions have been proposed [22] to allow to asses voice quality as a function of the transmission rate (see also [5]). Our findings in this second part can be summarized as follows:

- With a linear utility function, 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.
- In the case when all flows add FEC, which is 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 Section 2, we present the analysis and results for the first part. The analysis and results for the second part come in Section 3. We conclude the paper in Section 4 where we also briefly mention the case of general network topology and distributions of packet sizes and interarrival times. Note that although we are focusing on audio flows, our results on FEC are valid for any other kind of multimedia application.

2 Analysis of expected goodput

This part is organized as follows. In Subsection 2.1, we describe our general model for applications using FEC, and we define a quality function which we will use in the rest of this part. We then present the scenario of fixed amount of useful information per packet, and fixed packet size, in Subsection 2.2 and 2.3, respectively. In Subsection 2.4, we study the simple case when packet n carries redundant information of packet $n-1$ assuming an $M/M/1/K$ queuing model. In Subsection 2.5, we solve the problem for the general case when packet n carries redundant information of packet $n-\phi$, with $\phi \geq 1$. Finally, we look in Subsection 2.7 at the quality in the case of infinite spacing $\phi \to \infty$. We use this result to infer about multiplexing between several flows in Subsection 2.8.

2.1 Analysis

In a large network as the Internet, a flow of packets crosses several routers before reaching the other end. Most of the losses from a flow occur in the router having the smallest available bandwidth in the chain of routers, so that one may model the whole chain by one single router called "*the bottleneck*." This assumption has both theoretical and experimental justification [2,6]. We shall use the simple $M/M/1/K$ queue to model the network and thus the loss process of audio packets. In other words, we assume that audio packets arrive at the bottleneck according to a Poisson process of intensity λ, and we assume that the time required to process an audio packet at the bottleneck is exponentially distributed with parameter μ. The Poisson assumption on inter–arrival times could be justified by the random delay added to packets by routers located upstream the bottleneck. The service time represents the time between the beginning of the transmission of an audio packet on the bottleneck interface leading to the destination until the beginning of the transmission of the next packet from the same audio flow. Since the two packets may be spaced apart by a random number of packets from other applications, one may use the exponential distribution as a candidate for modeling the service time of audio packets at the bottleneck. The reason for choosing this simplistic model for the network is to be able to obtain simple mathematical formulas that give us some insights on the gain from using FEC.

Let $\rho = \lambda/\mu$ be the intensity of audio traffic. Assume that audio packets share alone the buffer K. This can be the case of a bottleneck crossed only by audio packets, or the case of a bottleneck router implementing a per–flow or a per–class queuing. Thus, for $\rho < 1$, the loss probability of an audio packet in steady state is given by [11]:

$$\pi(\rho) = \frac{1-\rho}{1-\rho^{K+1}}\rho^{K}, \tag{1}$$

and for $\rho = 1$ it is equal to

$$\pi(\rho) = \frac{1}{K+1}.$$

Now, we add redundancy to each packet in a way that if a packet is lost, it can be still "partially" retrieved if the packet containing its redundancy is not lost. The redundancy is located ϕ packets apart from the original packet. It consists in a low quality copy of the original packet. Let α be the ratio of the volume of the redundant information and the volume of the original packet. α is generally less than one. Along with the possibility to retrieve the lost information in the network, we should consider the negative impact of the addition of FEC. Two scenarios should be considered.

- The first, in which the amount of useful information in a packet does not change when adding FEC. In that case, the FEC has a negative impact on the loss probability. Indeed, additional redundant information has an impact on the service times since packets require now more time to be retransmitted at the output of the bottleneck. It may also have an impact on the buffering capacity at the bottleneck since each packet now contains more bits.
- In the second scenario, the packet size does not depend on the amount of added FEC. This means that the amount of useful information in a packet reduces in order to leave space for redundant information of a previous packet.

2.2 Constant amount of useful information in a packet

We shall propose the following two possible negative impacts of FEC, in order to study later the tradeoff between the positive and negative impacts:

- **Impact of FEC on service time.** We assume that audio packets including redundancy require a longer service time which is exponentially distributed with parameter $\frac{\mu}{(1+\alpha)}$. This can be the case when our audio flow has an important share of the bottleneck bandwidth. If it is not the case, this assumption can hold when the exogenous traffic at the bottleneck (or at least an important part of it) is formed of audio flows that implement the same FEC scheme. Our assumption also holds when the bottleneck router implements a per–flow scheduling that accounts for the size of packets.

- **Impact of FEC on buffering.** The buffering capacity in the bottleneck router will be affected by the addition of FEC in one of two ways: (1) Since packets are now longer by a factor $(1+\alpha)$, we can consider that the amount of buffering is diminished by this quantity, or (2) We can assume that the queue capacity is not function of packet length, but rather of the number of packets. Hence, the queue capacity is not affected by the use of FEC. Let K_α denote the buffer size after the addition of FEC in terms of packets. It is equal to $K/(1+\alpha)$ if the buffer capacity is changed, and it is equal to K otherwise. Thus, the loss probability in the presence of FEC takes the following form:

$$\pi_\rho(\alpha) = \frac{1-\rho(1+\alpha)}{1-\Big(\rho(1+\alpha)\Big)^{K_\alpha}}\Big(\rho(1+\alpha)\Big)^{K_\alpha}. \tag{2}$$

Before we define the quality of audio received at the destination, we introduce a random variable Y_n that indicates a successful arrival of a packet at the destination or not. Then,

$$\begin{aligned} Y_n &= 0, \quad \text{if packet } n \text{ is lost, and} \\ Y_n &= 1, \quad \text{if packet } n \text{ is correctly received.} \end{aligned}$$

Let $\phi \geq 1$ be the variable indicating the distance, or the offset, between the original packet and its redundancy. We make the simple assumption that the audio quality is proportional to the amount of information we receive. A quality equal to 1 indicates that we are receiving all the information (the basic audio flow). The quality we get after the reconstruction of an original packet from the redundancy is taken equal to α, where α is the ratio of redundancy volume and original packet volume. We thus define the quality function as,

$$\begin{aligned} Q(\alpha) &= P(Y_n = 1) + \alpha P(Y_n = 0) P(Y_{n+\phi} = 1 | Y_n = 0) \\ &= 1 - \pi_\rho(\alpha)(1 - \alpha P(Y_{n+\phi} = 1 | Y_n = 0)) \,. \end{aligned} \tag{3}$$

This equation gives us the audio quality at the destination under a FEC scheme of rate $(1+\alpha)^{-1}$, and of distance ϕ between an original packet and its redundancy. For the case $\alpha = 0$, our definition for the quality coincides with the probability that a packet is correctly received. For the case $\alpha = 1$, it coincides with the probability that the information in an original packet is correctly received, either because it was not lost, or because it was fully retrieved from the redundancy. One may imagine to use another quality function that the one we chose. In particular, one can use a quality function that is not only a function of the amount of data correctly received but also of the coding algorithm used. Different algorithms have been used in [9,17] for coding the original data and the redundancy. Table 1 resumes the notation we will use in the rest of the paper.

Expression	Definition
$Q(\alpha)$	The audio quality.
ϕ	The offset between the original packet and the packet including its redundancy.
K_α	The size of the queue.
X_j	The random variable which represents the number of packets in the queue just before the arrival of the j–th audio packet.
Z_j	The random variable which represents the number of services between the arrivals of the $j-1$–th and the j–th audio packets. [4]

Table 1
Notation used in this paper.

2.3 The case of constant packet size

We assume that the total packet size does not depend on the amount of FEC. A packet is constituted of a fraction α of redundant information, and a fraction of $1-\alpha$ of useful information.

Since the packet size is constant here, the FEC has no impact on the loss probabilities nor on the buffer size (in units of packets). In particular, the probability $\pi_\rho(\alpha)$ does not depend on α, and ρ neither does not change with α. So we do not need to include α and ρ in the notation.

The quality we get after the reconstruction of an original packet from the redundancy is taken equal to α. But if we do not lose the original packet,

[4] As is frequently done, we include in Z_j not only real services but also “potential services”: these are services that occur while the system is empty; thus at the end of such a service no packet leaves.

its quality is $1-\alpha$, instead of 1 unit, as before. We thus define the quality function as,

$$\begin{aligned} Q(\alpha) &= (1-\alpha)P(Y_n=1) + \alpha P(Y_n=0)P(Y_{n+\phi}=1|Y_n=0) \\ &= (1-\pi)(1-\alpha) + \alpha\pi P(Y_{n+\phi}=1|Y_n=0))\,. \end{aligned} \tag{4}$$

For both scenarios (where useful information or where total information in a packet are constant), we ask the following question: "*How does the audio quality vary as a function of α?*" That would permit us to evaluate the benefits from such a recovery mechanism and to find the appropriate amount of redundancy α that must be added to each packet. In the next sections we find the audio quality for different values of ϕ. The only missing parameter is the probability that the redundant information on a packet is correctly received given that the packet itself is lost. This is the function $P(Y_{n+\phi}=1|Y_n=0)$ in (3). In the following sections we put ourselves in the stationary regime and we compute this probability.

2.4 Spacing by $\phi = 1$

In this section we analyze the case when the redundant information of packet n is carried by packet $n+1$, i.e., $\phi=1$. This mechanism is implemented in well known audio tools as Freephone [9] and Rat [17]. Let R be the event that the redundancy is correctly received given that the original packet is lost. This represents the case where the next event after the loss of the original packet is a departure and not an arrival.

2.4.0.1 Fixed amount of useful information per packet. Consider first the scenario in which the useful information in a packet is fixed. Then the probability of event R is given by

$$P(Y_{n+1}=1|Y_n=0) = \frac{1}{\rho(1+\alpha)+1}. \tag{5}$$

Substituting (5) in (3), we obtain

$$Q_{\phi=1}(\alpha) = 1 - \pi_\rho(\alpha)\Big(1 - \frac{\alpha}{\rho(1+\alpha)+1}\Big).$$

To study the impact of FEC on the audio quality, we plot $Q_{\phi=1}(\alpha)$ as a function of α for different values of K_α and ρ. In Figure 2 we show the results when the buffering capacity at the bottleneck is assumed to change with the amount of FEC ($K_\alpha = K/(1+\alpha)$), and in Figure 3 we show the results for the case where the buffering capacity is not changed ($K_\alpha = K$). We see that, for both cases, audio quality deteriorates when α increases (when we add more redundancy), and this deterioration becomes more important when the traffic intensity increases and when the buffer size decreases. The main interpretation of such behavior is that the loss probability of an original packet increases with α faster than the gain in quality we got from retrieving the redundant information. This should not be surprising. Indeed, even in more sophisticated schemes in which a single redundant packet is added to protect a whole block of M packets, it is known that FEC often has an overall negative effect, see [7,1,10]. Yet in such schemes the negative effect of adding the redundancy is smaller than in our scheme, since the amount of added information per packet is smaller (i.e., a single packet protects a whole group of M packets). But, we know that for such schemes and in case of light traffic, the overall contribution of FEC is positive [1,10]. This motivates us to analyze more precisely the impact of FEC in our simplistic scheme in case of light traffic.

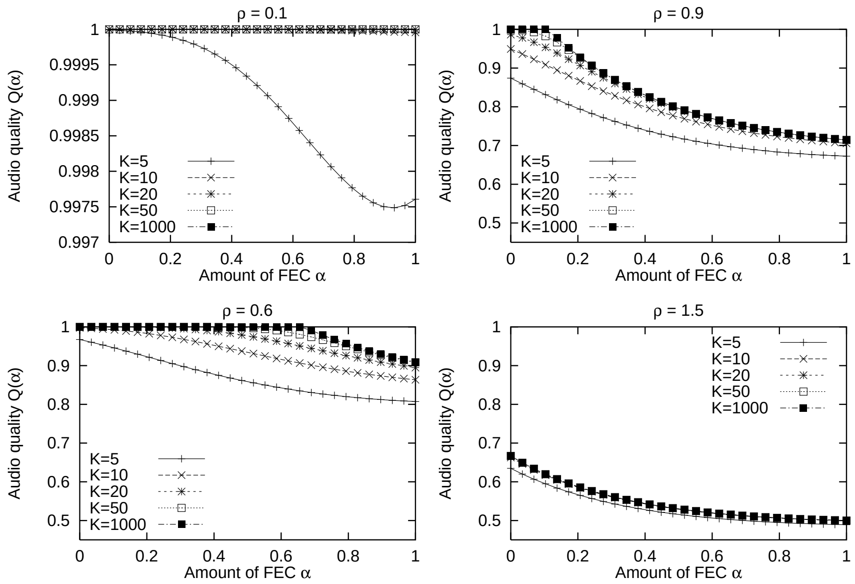


Fig. 2. $\phi = 1$ and the queue capacity is changed.

Define the function $\Delta(\rho) = Q(1) - Q(0)$ and consider the case when the buffering capacity at the bottleneck is not affected by the amount of FEC. This is an optimistic scenario where it is very probable to see the gain brought by FEC, of course if this gain exists. We have,

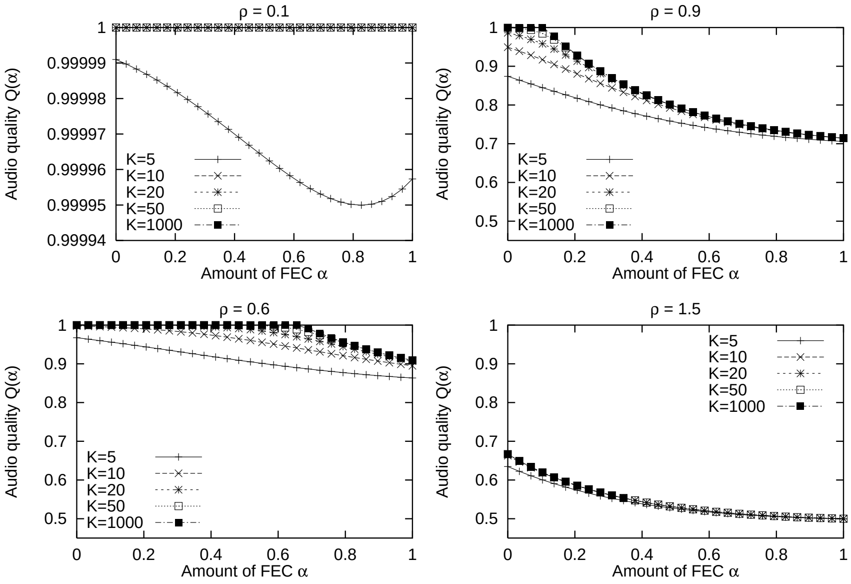


Fig. 3. $\phi = 1$ and the queue capacity is not changed.

$$\Delta(\rho) = -2(2\rho)^{\frac{K_\alpha}{2}} \left(\frac{1+\rho}{2\rho+1}\right)\left(\frac{1-2\rho}{1-(2\rho)^{\frac{K_\alpha}{2}+1}}\right) + \frac{1-\rho}{1-\rho^{K_\alpha+1}}\rho^{K_\alpha} \tag{6}$$

Finding $\lim_{\rho\to 0}\Delta(\rho)$ would permit us to evaluate the audio quality for a very low traffic intensity. We took $K_\alpha = 2M$ in (6) and we expanded $\Delta(\rho)$ in a Taylor series. We found that all the first coefficients of the series $c_0, c_1, ..., c_{M-1}$ are equal to zero, and that the coefficient c_M is negative and equal to $-2(2\rho)^M$. c_i is the coefficient of ρ^i in the Taylor series of $\Delta(\rho)$ and can be computed by

$$c_j = \frac{d}{d\rho^j}\Delta^j(\rho)_{|\rho=0}.$$

Thus, for small ρ, $\Delta(\rho)$ can be written as $-2(2\rho)^M + o(\rho^M)$ and the gain from the addition of FEC can be seen to be negative. With this simple FEC scheme, we lose in audio quality when adding FEC even for a very low traffic intensity. This loss in quality decreases with the increase in buffer size.

2.4.0.2 Fixed packet size. Substituting (5) in (4), we obtain

$$Q_{\phi=1}(\alpha) = (1-\alpha)(1-\pi) + \frac{\alpha\pi}{\rho(1+\alpha)+1}.$$

In Section 2.7 we shall show that we loose in that case, as well as for any general spacing $\phi > 1$. This will be done by considering an optimistic bound obtained by an infinite spacing.

2.5 General case: Spacing by $\phi \geq 1$

Now, we consider the more general case when the spacing between the original packet and its redundancy is greater than 1. The idea behind this type of spacing is that losses in real networks tend to appear in bursts, and thus spacing the redundancy from the original packet by more than one improves the probability to retrieve the redundancy in case the original packet is lost. Indeed, a packet loss means that the queue is full and thus the probability of losing the next packet is higher than the steady state probability of losing a packet. The spacing gives the redundancy of a packet more chance to find a non full buffer at the bottleneck, and thus to be correctly received. We note that the phenomenon of the correlation between losses of packets was already modeled and studied in other papers: [7,1,10]. Measurements have also shown that most of the losses are correlated [4,3,20].

Here, we are interested in finding the probability that packet $n + \phi$ is lost given that packet n is also lost. This will give us $P(Y_{n+\phi} = 1|Y_n = 0)$ which in turn gives us the expression for the audio quality (as expressed in (3)). Since we assume that the system is in its steady state, we can omit the index n and substitute it by zero. We have $Y_0 = 0$ which means $X_0 = K_\alpha$. We are interested in the probability that $X_\phi = K_\alpha$. For the ease of computation, we consider the case $\phi \leq K_\alpha$. We believe that this is quite enough given that a large spacing between the original packet and the redundancy leads to an important jitter and a poor interactivity.

In order to obtain an explicit expression for the probability $P(X_\phi = K_\alpha|X_0 = K_\alpha)$, we first provide an explicit sample–path expression for the event of loss of the packet carrying the redundancy, given that the original packet itself was lost.

Theorem 1 *Let $X_0 = K_\alpha$ and $1 \leq \phi \leq K_\alpha$. then:*

Packet ϕ is not lost if and only if

$$X_\phi < K_\alpha \Leftrightarrow \begin{cases} Z_\phi - 1 \geq 0 \\ \quad or \\ Z_\phi + Z_{\phi-1} - 2 \geq 0 \\ \quad or \\ \quad \vdots \\ \quad or \\ Z_\phi + Z_{\phi-1} + \cdots + Z_1 - \phi \geq 0 \end{cases} \tag{7}$$

or equivalently, packet ϕ is lost if and only if

$$X_\phi = K_\alpha \Leftrightarrow \begin{cases} Z_\phi - 1 < 0 \\ \quad and \\ Z_\phi + Z_{\phi-1} - 2 < 0 \\ \quad and \\ \quad \vdots \\ \quad and \\ Z_\phi + Z_{\phi-1} + \cdots + Z_1 - \phi < 0 \end{cases} \tag{8}$$

Proof: We can express the number of packets that the $i+1$–th audio packet will find in the queue upon arrival as follows:

$$X_{i+1} = \Big((X_i + 1) \wedge K_\alpha - Z_{i+1}\Big) \vee 0 \quad \forall\, i \geq 0, \tag{9}$$

where $\wedge$ and $\vee$ are respectively the minimum and maximum operators. The rest of the proof goes in three steps that are summarized in Lemma 1, Lemma 2 and Corollary 1 below. $\diamond$

Now, we define

$$\tilde{X}_{i+1} \triangleq (\tilde{X}_i + 1) \wedge K_\alpha - Z_{i+1}. \tag{10}$$

This new variable corresponds to the number of packets that would be found in the queue upon the arrival of packet $i+1$ if the queue size could become negative. We next show that it can be used as a lower bound for X_{i+1}.

Lemma 1 *If $\tilde{X}_0 \leq X_0$ then $\tilde{X}_i \leq X_i \quad \forall i \geq 0$.*

Proof: We proceed for the proof by induction. This relation is valid for $i = 0$. Suppose that it is valid for $i \geq 0$. We show that it is valid for $i + 1$,

$$\begin{aligned}
\tilde{X}_{i+1} &\leq \Big((\tilde{X}_i + 1) \wedge K_\alpha - Z_{i+1} \Big) \vee 0 \\
&\leq \Big((X_i + 1) \wedge K_\alpha - Z_{i+1} \Big) \vee 0 \\
&= X_{i+1}.
\end{aligned}$$

$\diamond$

Lemma 2 *Let $\tilde{X}_0 = K_\alpha$, then*

$$\tilde{X}_i = K_\alpha - \max_{1 \leq l \leq i} \textstyle\sum_{j=l}^{i} (Z_j - 1) - 1 \qquad \forall i \geq 0.$$

Proof:

$$\begin{aligned}
\tilde{X}_0 &= K_\alpha \\
\tilde{X}_1 &= K_\alpha - Z_1 \\
\tilde{X}_2 &= (K_\alpha - Z_1 + 1) \wedge K_\alpha - Z_2 \\
\tilde{X}_3 &= \Big((K_\alpha - Z_1 + 1) \wedge K_\alpha - Z_2 + 1 \Big) \wedge K_\alpha - Z_3 \\
&= (K_\alpha - Z_1 - Z_2 + 2) \wedge (K_\alpha - Z_2 + 1) \wedge K_\alpha - Z_3 \\
&= K_\alpha - (Z_1 + Z_2 - 2) \vee (Z_2 - 1) \vee 0 - Z_3 \\
&\vdots \\
\tilde{X}_i &= K_\alpha - \max_{1 \leq l < i} \Big\{ 0, \sum_{j=l}^{i-1} (Z_j - 1) \Big\} - Z_i \\
&= K_\alpha - \max_{1 \leq l < i} \Big\{ 0, \sum_{j=l}^{i-1} (Z_j - 1) \Big\} - Z_i \\
\Rightarrow \tilde{X}_i &= K_\alpha - \max_{1 \leq l \leq i} \Big\{ \sum_{j=l}^{i} (Z_j - 1) \Big\} - 1.
\end{aligned}$$

$\diamond$

Corollary 1 *Expression (8) holds if $X_0 = K_\alpha$ and $\phi \leq K_\alpha$.*

Proof: The right hand side in (8) is no other than:

$$\max_{1 \leq l \leq \phi} \Big\{ \sum_{j=l}^{\phi} (Z_j - 1) \Big\}.$$

Suppose first that $\max_{1\le l\le\phi}\Big\{\sum_{j=l}^{\phi}(Z_j-1)\Big\}<0$. Using Lemma 2 then Lemma 1, we have $\tilde{X}_\phi \ge K_\alpha$ which gives $X_\phi \ge K_\alpha$. Thus, $X_\phi = K_\alpha$.

Now, we need to show that if $X_\phi = K_\alpha$ we get

$$\max_{1\le l\le\phi}\Big\{\sum_{j=l}^{\phi}(Z_j-1)\Big\}<0.$$

We define:

$$\phi^* = \min\Big\{i \quad | \quad \tilde{X}_i < X_i\Big\}. \tag{11}$$

According to (11), we distinguish between the two following cases:

- $\phi^* > \phi$, and
- $\phi^* \le \phi$.

Consider the first case. Using the definition of ϕ^* and Lemma 2, we write: $\phi^* > \phi \quad \Rightarrow \quad \tilde{X}_\phi = X_\phi \Rightarrow \tilde{X}_\phi = K_\alpha \Rightarrow \max_{1\le l\le\phi}\Big\{\sum_{j=l}^{\phi}(Z_j-1)\Big\} = -1 < 0.$

Now, suppose that $\phi^* \le \phi$, thus $\tilde{X}_{\phi^*} < 0$ and $X_{\phi^*} = 0$. We write,

$$X_\phi \le X_{\phi^*} + (\phi-\phi^*) = (\phi-\phi^*) < \phi \le K_\alpha,$$

if there were no service. Thus, we get in this case $X_\phi < K_\alpha$ which is in contradiction with our assumption that $X_\phi = K_\alpha$. The case $\phi^* \le \phi$ does not appear if ϕ is chosen less or equal to the buffering capacity. Thus, for $X_\phi = K_\alpha$ we have $\max_{1\le l\le\phi}\Big\{\sum_{j=l}^{\phi}(Z_j-1)\Big\}<0$. This concludes the proof of Theorem 1. $\diamond$

According to Ballot's Theorem [23] (see the Appendix in Section 4 for details), we have for $k < \phi$:

Lemma 3

$$P\Big\{\max_{1\le l\le\phi}\Big\{\sum_{j=l}^{\phi}(Z_j-1)\Big\}<0 \,|\, \sum_{l=1}^{\phi} Z_l = k\Big\} = 1-\frac{k}{\phi}. \tag{12}$$

Let A be the event that $X_\phi = K_\alpha$ given that $X_0 = K_\alpha$. We sometimes write A^ϕ to stress the dependence on ϕ. We conclude from Theorem 1 that if packet 0 is lost, i.e. if packet 0 finds K_α packets in the system, then

$$A = \left\{ \max_{1 \le l \le \phi} \left\{ \sum_{j=l}^{\phi} (Z_j - 1) \right\} < 0 \right\}.$$

Then, we can represent the probability that packet $n + \phi$ is lost given that packet n is lost as

$$P(Y_{n+\phi} = 0 | Y_n = 0) = P(A)$$

$$= \sum_{k=0}^{\phi-1} P(A | Z_1 + \cdots + Z_\phi = k) P(Z_1 + \cdots + Z_\phi = k) \tag{13}$$

Once this probability is computed, the audio quality can be directly derived using (3).

Theorem 2 *Define $\rho_\alpha = \rho(1+\alpha)$ and for the scenario in which the useful amount of information in a packet is constant in α; define $\rho_\alpha = \rho$ and $K_\alpha = K$ in the scenario in which the packet size does not depend of α. Consider $1 \le \phi \le K_\alpha$. Given that packet n is lost, the probability that packet $n + \phi$ is also lost is given by*

$$P(A) = \sum_{k=0}^{\phi-1} \left(1 - \frac{k}{\phi}\right) \left(\frac{\rho_\alpha}{\rho_\alpha + 1}\right)^{\phi} \left(\frac{1}{\rho_\alpha + 1}\right)^{k} \binom{\phi + k - 1}{\phi - 1}, \tag{14}$$

where $\binom{n}{k} = \frac{n!}{(n-k)!k!}$ denotes the binomial coefficient. The quality function can be computed by substituting $P(A)$ in (3). Note that $P(Y_{n+\phi} = 1 | Y_n = 0) = 1 - P(A)$.

Proof: The second right hand term of (13) must be solved by combinatorial reasoning. For that purpose, we define the vector $\vec{Z}$ to be:

$$\vec{Z} = \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_\phi \end{pmatrix}, \tag{15}$$

where $\sum_{l=1}^{\phi} Z_l = k$, and we define S be the set of the different sets that $\vec{Z}$ may acquire: $S = \{\vec{Z}\}$. We must sum over all the possible trajectories:

$$
\begin{aligned}
P(\sum_{l=1}^{\phi} Z_l = k) &= \sum_{S} P(Z_1 = z_1)P(Z_2 = z_2)\cdots P(Z_\phi = z_\phi) \\
&= \sum_{S} \Big(\frac{\lambda}{\lambda+\mu_\alpha}\Big)^{\phi}\Big(\frac{\mu_\alpha}{\lambda+\mu_\alpha}\Big)^{k} \\
&= \Big(\frac{\lambda}{\lambda+\mu_\alpha}\Big)^{\phi}\Big(\frac{\mu_\alpha}{\lambda+\mu_\alpha}\Big)^{k}\binom{\phi+k-1}{\phi-1}. \qquad (16)
\end{aligned}
$$

We define here μ_α as being equal to $\mu/(1+\alpha)$ for the scenario in which the amount of useful information per packet does not depend on α, and $\mu_\alpha = \mu$ in the case of fixed packet size. It's easy to see that the combinatorial part of (16) holds. To do that, we can see the problem to be the number of distinguishable arrangements of k indistinguishable objects (the packet audio departures from the bottleneck) in ϕ inter–arrival intervals, just as it's depicted in Figure 4.

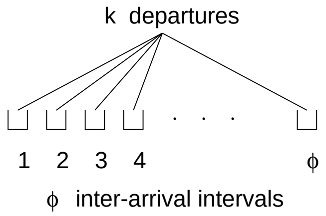


Fig. 4. Model to solve the combinatorial part.

Using (16) we get finally,

$$
P(A) = \sum_{k=0}^{\phi-1}\Big(1-\frac{k}{\phi}\Big)\Big(\frac{\lambda}{\lambda+\mu_\alpha}\Big)^{\phi}\Big(\frac{\mu_\alpha}{\lambda+\mu_\alpha}\Big)^{k}\binom{\phi+k-1}{\phi-1}, \qquad (17)
$$

which yields (14) in terms of $\rho_\alpha = \rho(1+\alpha) = \lambda/\mu_\alpha$. The quality function can be obtained by substituting (14) in (3). The value of $\pi_\alpha(\rho)$ is given in (2). $\diamond$

We trace now plots of the audio quality as given by (3) and (14) for different values of K_α, ϕ and ρ. Figure 5 depicts the behavior of $Q(\alpha)$ when the buffering capacity at the bottleneck is assumed to be divided by a factor $(1+\alpha)$, and Figure 6 depicts this behavior when the buffering capacity is not changed.

We notice that, just as in the case of $\phi = 1$, we always lose in quality when we increase the amount of FEC even if we consider a large spacing. But, we also notice that for a given amount of FEC, the quality improves when spacing the redundancy from the original packet. This is the result of an improvement in the probability to retrieve the redundancy given that the original packet is lost. This monotonicity property holds, in fact, for any value of ϕ (not just for $\phi \le K_\alpha$). We show this theoretically in the next section.

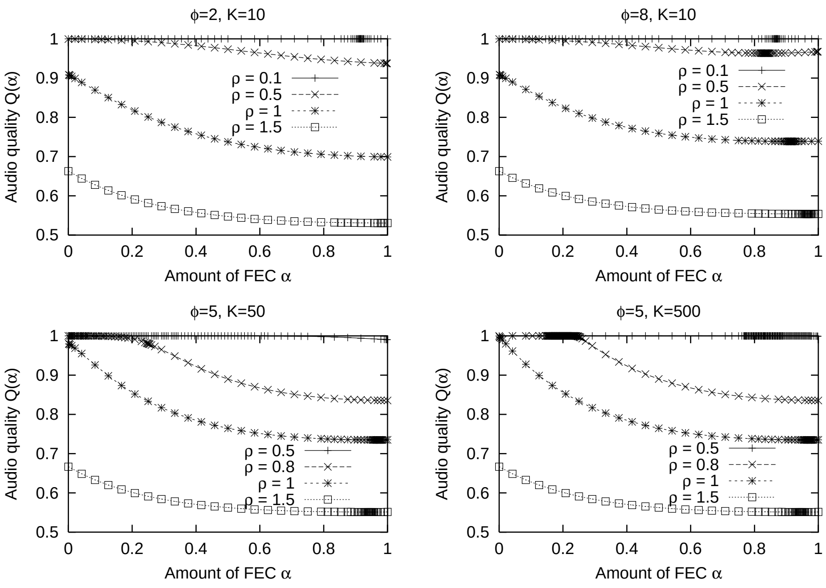


Fig. 5. Quality behavior in the presence of FEC and spacing $1 < \phi < K_\alpha$ assuming that queue size is changed.

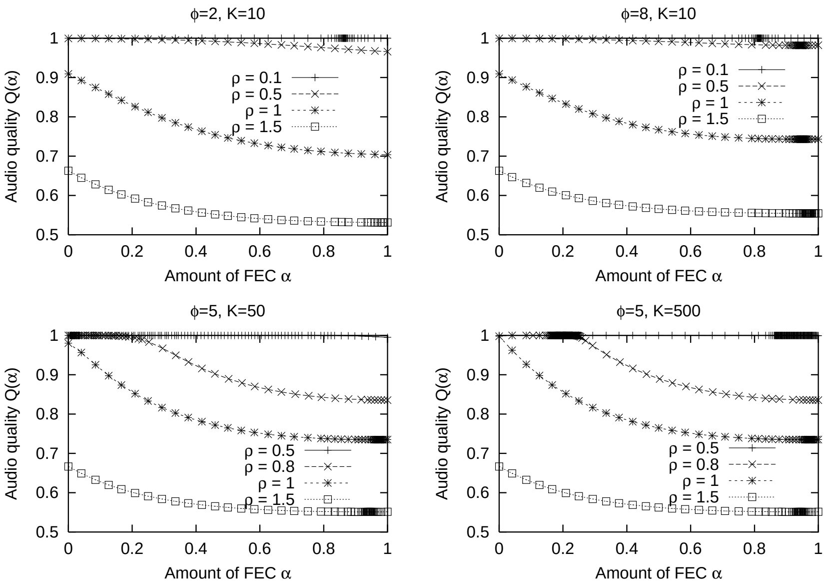


Fig. 6. Quality behavior in the presence of FEC and spacing $1 < \phi < K_\alpha$ assuming that queue size is not changed.

2.6 Monotone increase of the quality with the spacing

The steady state probability of loss of a packet n does not depend on ϕ. It thus remains to check the behavior of $P(X_{n+\phi} = K_\alpha | X_n = K_\alpha)$ as a function of ϕ in order to decide on the quality variation (Eq. 3). The quality is a decreasing function of this probability. For $\phi \leq K_\alpha$, the latter probability is equal to $P(A^\phi)$, and the monotonicity property can be seen directly from the fact that

A^ϕ is a monotone decreasing set (since it requires for more summands to be smaller than zero, as ϕ increases, see Eq. 8).

Now, to see that $P(X_{n+\phi} = K_\alpha | X_n = K_\alpha)$ is monotone decreasing for any ϕ, we observe (9), which holds for any $i > 0$, and note that X_{i+1} is monotone increasing in X_i. Thus by iteration, we get that X_ϕ is monotone increasing in X_0. Now using this monotonicity, we have

$$\begin{aligned} P(X_{\phi+1} = K_\alpha | X_0 = K_\alpha) &= P(X_\phi = K_\alpha | X_{-1} = K_\alpha) \\ &= \sum_{i=0}^{K_\alpha} P(X_\phi = K_\alpha | X_0 = i, X_{-1} = K_\alpha) \times P(X_0 = i | X_{-1} = K_\alpha) \end{aligned}$$

$$\begin{aligned} &= \sum_{i=0}^{K_\alpha} P(X_\phi = K_\alpha | X_0 = i) P(X_0 = i | X_{-1} = K_\alpha) \\ &\leq \sum_{i=0}^{K_\alpha} P(X_\phi = K_\alpha | X_0 = K_\alpha) P(X_0 = i | X_{-1} = K_\alpha) \\ &= P(X_\phi = K_\alpha | X_0 = K_\alpha). \end{aligned}$$

2.7 Limiting case: Spacing $\phi \to \infty$

The case of large ϕ is not of interest in interactive applications, since it means unacceptable delay. However, since we have found that the quality of the audio with FEC improves as the spacing grows, it is natural to study the limit ($\phi \to \infty$) in order to get an upper bound. Indeed, if we see that in this limiting case we do not improve the quality, it means that we lose by adding FEC according to our simple scheme for any finite offset ϕ.

When $\phi \to \infty$, the probability that the redundancy is dropped becomes equal to the steady state drop probability of a packet.

Consider the scenario of fixed amount of useful information per packet. Then, (3) can be written as,

$$Q_{\phi\to\infty}(\alpha) = 1 - \pi_\rho(\alpha) + \alpha\pi_\rho(\alpha)(1 - \pi_\rho(\alpha)) \,. \tag{18}$$

We plot (18) in Figure 7 as a function of the amount of FEC for different values of K_α and ρ. We see well how, although we are in the most optimistic case, we lose in quality when adding FEC. That suggests that this class of

FEC mechanisms are not efficient for real time transmission because it never improves the goodput of the connection.

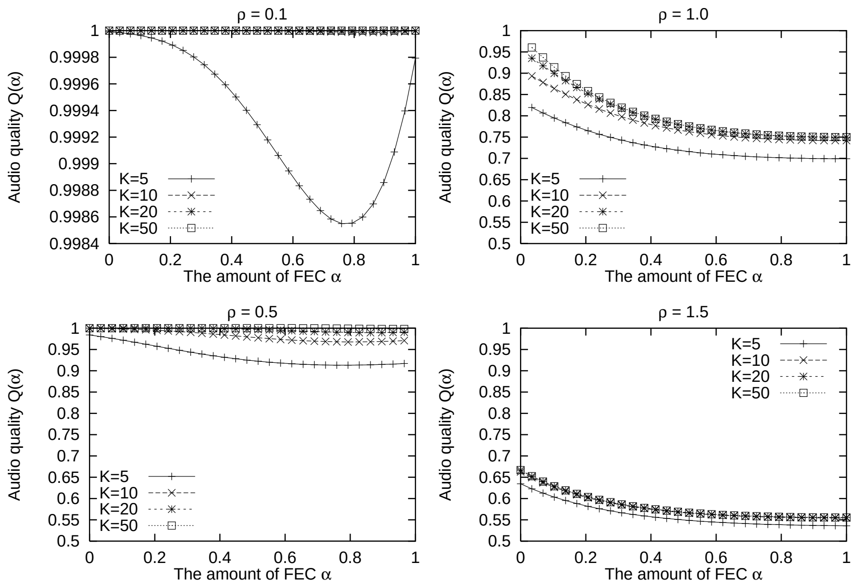


Fig. 7. Quality behavior in the presence of FEC and spacing $\phi \to \infty$.

Next we show that this conclusion also holds for the scenario of fixed packet size. In the case of very large spacing, (4) can be written as

$$Q_{\phi\to\infty}(\alpha) = (1-\pi)(1-\alpha) + \alpha\pi(1-\pi) = (1-\pi)(1-\alpha(1-\pi))\,. \qquad (19)$$

We see that this is strictly smaller than $1-\pi$ which is the quality with zero redundancy!

2.8 Multiplexing between traffic

We analyze the case when several input flows arrive to the bottleneck, an audio flow and an exogenous flow which represent the superposition of all other flows. We further consider here the case of general independent service time distribution. We use the results of the previous subsection to show that we do not gain in goodput by multiplexing (under both FEC scenarios) provided that the packet size of all multiplexed streams has the same distribution. This covers both the first scenario (in which packet size increases with FEC) where all streams add FEC, as well as the second scenario in which it does not matter any more how many streams add FEC (since the packet size is not affected by FEC).

We model the multiplexing by the arrival of two independent Poisson input flows, λ_1 and λ_2, representing the audio flow and the exogenous traffic (which represents all other flows) respectively.

In order to compute the goodput of a flow that adds FEC, we can use the fact that the spacing between a packet and its redundancy is now random, with a geometric distribution with parameter λ_1/λ_2. We note that we cannot use anymore the exact expressions for loss probabilities derived in Subsection 2.5, since the spacing can now be larger than the buffer size. However, we can still use the fact that the quality is increasing with the spacing. Thus, since we saw that we lose by adding FEC for $\phi \to \infty$, we conclude that we lose in goodput by adding FEC when multiplexing flows under the above assumptions. Nevertheless, we shall show in the next section that we may gain in goodput under different assumptions.

3 Cases where FEC improves audio quality

As we explained in the Introduction, the negative result in the first part of the paper 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 linear utility function we considered. In this part, we consider other cases where we prove that FEC may improve the audio quality. In Section 3.1, we investigate the case of a single audio flow sharing the bottleneck with an exogenous traffic not implementing FEC. In Section 3.2, we study the performance of FEC for other non–linear utility functions. The results of this part serve as guidelines for an efficient use of FEC in audio applications. They will give us an explanation of the gain in audio quality we may perceive in some real scenarios. Queuing models similar to those used in the first part of the paper will be used through the second part.

3.1 Multiplexing and FEC performance

3.1.1 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 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 [11,8]. 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.

3.1.2 *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 denotes 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 [8]. 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. It can also be computed using the theorem of residues as follows:

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

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 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

$$\rho_\alpha = \lambda \Big(\frac{\beta(1+\alpha)}{\mu_0} + \frac{(1-\beta)}{\mu} \Big) . \tag{20}$$

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). For simplicity, we assume in this part that the size of the bottleneck buffer in terms of packets does not change with the addition of FEC. 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

$$Q^\phi_\alpha = (1 - \pi_\alpha) + U(\alpha)\pi_\alpha(1 - \pi^\phi_\alpha) . \tag{21}$$

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^ϕ_α 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, and as in the first part, we will only consider the case of a linear utility function $U(\alpha) = \alpha$. We keep the study of the impact of other utility functions until Section 3.2.

The exact computation of Q_α^ϕ requires the computation of π_α^ϕ. This latter function is quite difficult to compute given the multiplexing of packets from both flows at the bottleneck. We must sum 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 Section 2.6, 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

$$Q_\alpha^0 = (1-\pi_\alpha) + \alpha\pi_\alpha(1-\pi_\alpha^0)\,, \tag{22}$$
$$Q_\alpha = (1-\pi_\alpha)(1+\alpha\pi_\alpha)\,. \tag{23}$$

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. The analysis in the first part has shown 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 3 *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 (20).

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 [11], 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 computation 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.

◇

3.1.3 Numerical results

We solve numerically the model for the two bounds on the audio quality (Eq. 22 and 23). 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 8 shows the results.

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.

3.2 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 the first part can be caused by 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 (21) 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 [22] 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

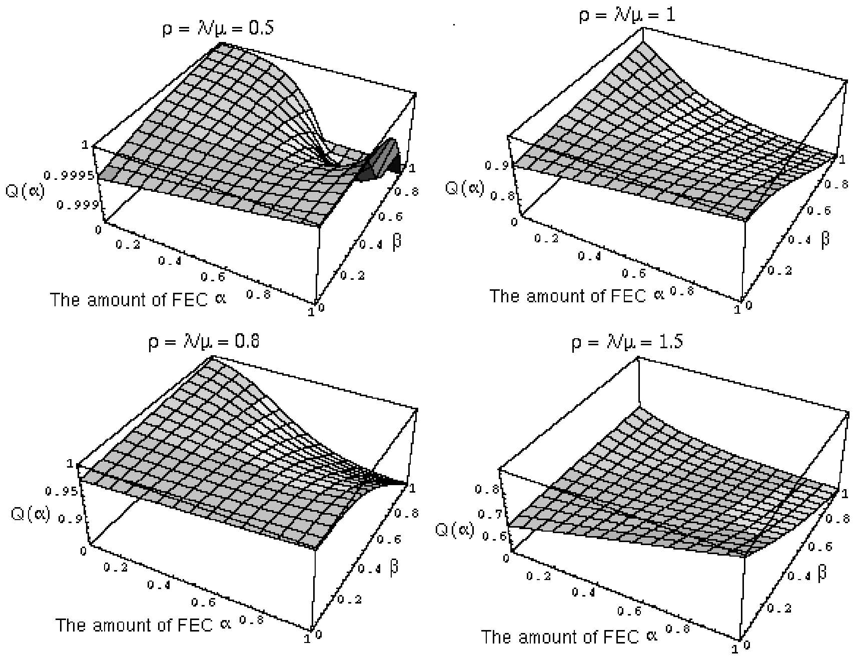


Fig. 8. Audio quality for an $M/G/1/K$ queue with two flows: the audio flow and the exogenous flow. β represents the probability that an arriving packet belongs to the audio flow. We see clearly how when $\beta \to 0$, Q_α^ϕ starts having an increasing behavior, and this gain becomes more important as ρ increases.

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 a quality close to the that of the original packet.

For the scenario of fixed amount of useful information per packet, 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 [21]. Analytically speaking, a utility function leads to an improvement of quality if for $\alpha < 1$, we have

$$Q_\alpha^\phi = (1 - \pi_\alpha) + U(\alpha)\pi_\alpha(1 - \pi_\alpha^\phi) > 1 - \pi_0\,,$$

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

In the scenario of fixed packet size (which does not depend on the amount of FEC), we gain by adding FEC if

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

(where we assumed $U(0) = 0, U(1) = 1$).

In the sequel we shall show how FEC can improve the quality for the scenario of fixed amount of useful information per packet. Similar improvement can be shown to occur also in the scenario of fixed packet size.

3.2.1 Some bounds on quality improvement

We already showed that we do not gain by multiplexing if the distribution of the sizes of all packets does not depend on the amount of FEC. We thus consider only the scenario in which the useful information is fixed per packet in the flow that adds FEC, and thus the packet size of this flow increases with FEC; we assume that other flows do not use FEC.

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

$$\begin{aligned} Q_\alpha^0 &= (1-\pi_\alpha) + U(\alpha)\pi_\alpha(1-\pi_\alpha^0)\,, \\ Q_\alpha &= (1-\pi_\alpha)(1+U(\alpha)\pi_\alpha) \end{aligned}$$

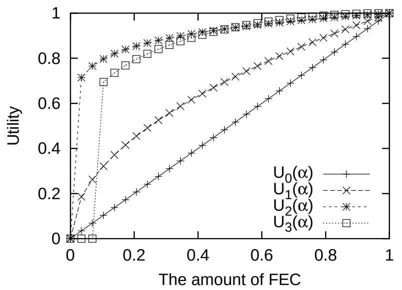
A utility function that improves the lower bound improves the quality for any value of ϕ. A utility function that does not improve the upper bound will not lead to an improvement of quality whatever is the value of ϕ. Using the upper bound, we can find the maximum quality that this simple FEC scheme 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 computation 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.

3.2.2 Some numerical results

We give in Figure 9 some possible utility functions [5] that could serve to our needs, and that are similar in their form to the utility functions proposed in [22]. In Figure 9, $U_3(\alpha)$ is plotted with $\alpha_0 = 0.1$.



$$U_0(\alpha) = \alpha$$

$$U_1(\alpha) = \sqrt{\alpha}$$

$$U_2(\alpha) = \alpha^{\frac{1}{10}}$$

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

Fig. 9. Possible utility functions for rate adaptive applications.

We solve the model numerically for the two bounds on the quality. We compute first the stationary distribution of the model for different values of α and ρ. We set K to 20 and λ to 10000 packets/sec. Then, for the the different utility functions in Figure 9, we plot the upper and lower bounds on the quality (Q_α and Q^0_α). Figure 10 shows plots for the lower bound and Figure 11 shows plots for the upper bound. The top four plots were obtained with $\alpha_0 = 0.1$ and the

[5] 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.

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.

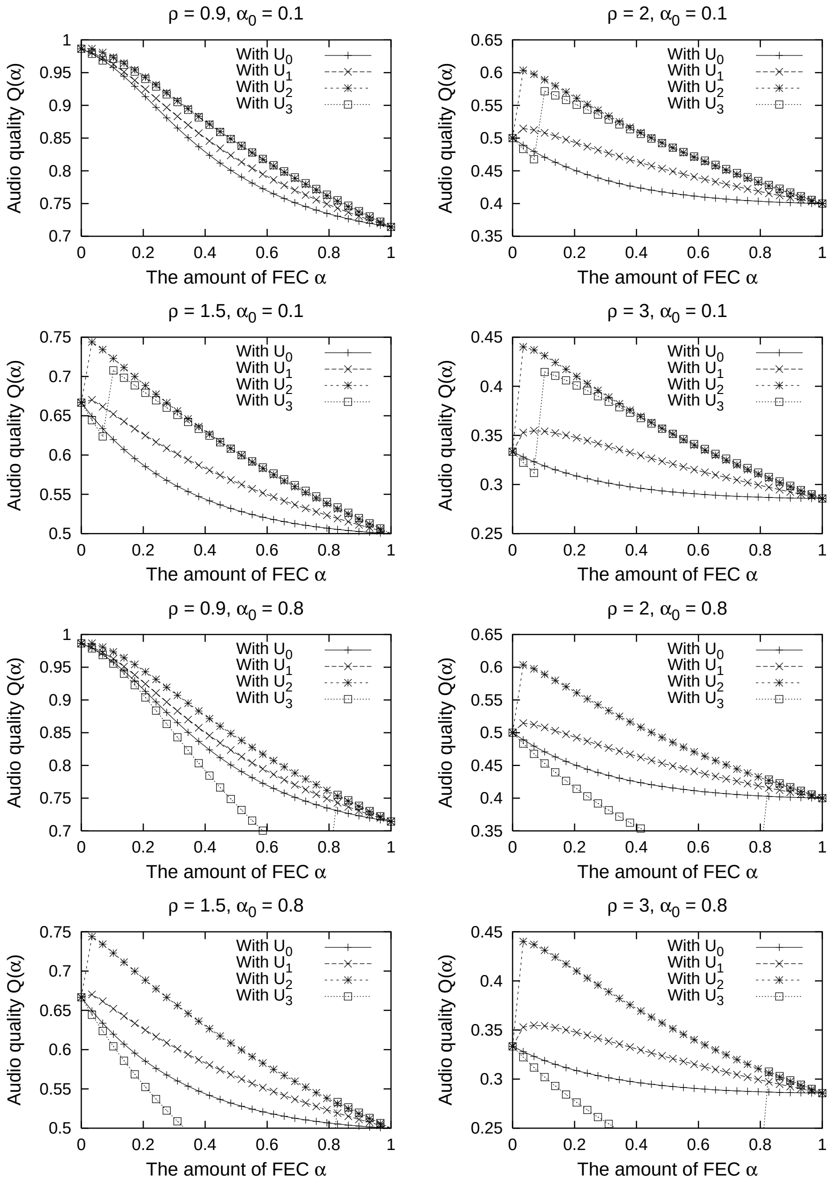


Fig. 10. Lower bound for audio quality with $K = 20$, $\alpha_0 = 0.1$ (top) and $\alpha_0 = 0.8$ (bottom).

4 Conclusions

We studied in this paper the effect on audio quality of a FEC scheme similar to the one used in [9,17]. This FEC scheme consists in adding a copy of an

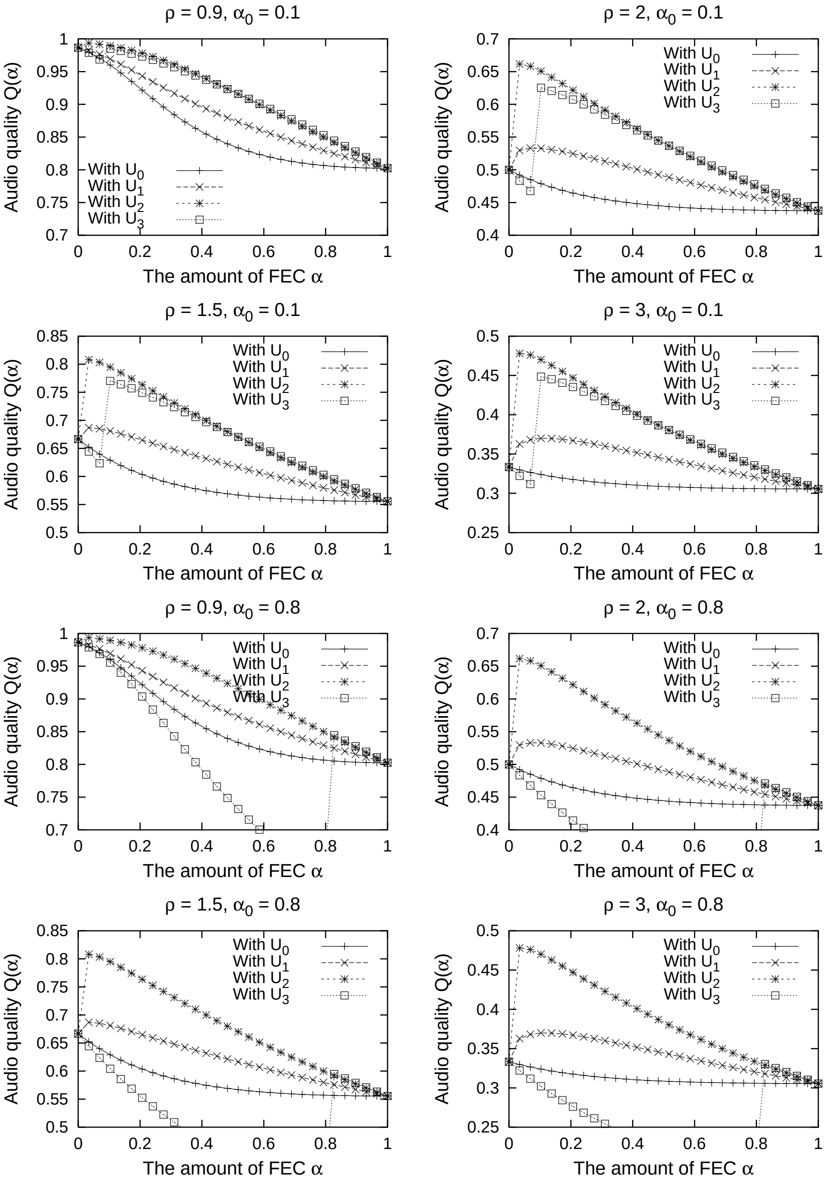


Fig. 11. Upper bound for audio quality with $K = 20$, $\alpha_0 = 0.1$ (top) and $\alpha_0 = 0.8$ (bottom).

audio packet to a subsequent packet so that the copy can be used when the original packet is lost. First, we considered the case when all flows in the network implements such a FEC scheme. With a simple queuing analysis and assuming linear utility functions, we found an explicit expression for the audio quality (corresponding to the goodput) for a general offset between an audio packet and its copy. This expression as well as some optimistic bound showed that the addition of FEC deteriorates the goodput instead of improving it.

One might wonder whether the conclusions depend on the probabilistic assumptions. However, one can note that the form of the optimistic bound (19) does not depend on the exponential assumptions: it would be the same for any service time and interarrival distributions, and even for topologies much more complex than a single queue. The precise values of π will of course depend on the distribution, and the form of the network, but the conclusion that we draw from the general form of (19) will remain the same.

We then studied cases where the FEC scheme may be helpful. The first case is when the audio flow has a small rate compared to an exogenous traffic that does not implement FEC. 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. We also gave an upper bound on the gain in audio quality we could obtain.

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 profits from the fact that most of the other flows do not implement FEC. This will not be the case when a large number of 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 especially when a small amount of redundancy gives the same performance as the big original packet. But it then seems intuitive to reduce the volume of original packets to reduce the drop probability and to gain even more in quality instead of adding FEC that does not improve the performance by 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 gave the receiver more choices by constructing at the source the redundancy carried by a packet from a block of audio packets. This will the topic of our future research in this direction.

A Ballot theorems

In this appendix, we cite the Ballot theorem that we have used to solve the problem for case $1 \le \phi \le K_\alpha$. The reader is referred to [23] for details.

Theorem 4 *Suppose that an urn contains n cards marked with nonnegative integers $k_1, k_2, \dots, k_n$, respectively, where $k_1 + k_2 + \cdots + k_n = k \le n$. All the n cards are drawn without replacement from the urn. Denote by ν_r, $r = 1, 2, \dots, n$, the number of the card drawn at the rth drawing. Then,*

$$P\{\nu_1 + \cdots + \nu_r < r \quad for \quad r = 1, \ldots, n\} = 1 - \frac{k}{n}, \tag{A.1}$$

provided that all possible results are equally probable.

References

[1] 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):779–787, 1998.

[2] J.-C. Bolot. End-to-end delay and loss behavior in the Internet. In *Proceedings of the ACM SIGCOMM*, pages 289–298, San Francisco, CA, Sept. 1993.

[3] J.-C. Bolot, H. Crépin, and A. V. García. Analysis of audio packet loss in the Internet. In *Proceedings od the International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, pages 163–174, Durham, New Hampshire, USA, Apr. 1995.

[4] J.-C. Bolot and A. V. García. Control mechanisms for packet audio in the Internet. In *Proceedings of the IEEE Infocom*, volume 1, pages 232–239, San Francisco, California, USA, Mar. 1996.

[5] C. Boutremans and J.-Y. L. Boudec. Adative delay aware error control for Internet telephony. In *Proceedings of the IP Telephony Workshop*, New York, USA, Apr. 2001.

[6] O. J. Boxma. Sojourn times in cyclic queues – the influence of the slowest server. In *Computer Performance and Reliability*, pages 13–24. Elsevier Science Publishers B.V. (North-Holland), 1988.

[7] I. Cidon, A. Khamisy, and M. Sidi. Analysis of packet loss process in high-speed networks. *IEEE Transactions on Information Theory*, IT-39(1):98–108, Jan. 1993.

[8] J. W. Cohen. *The Single Server Queue*. North-Holland, 1969.

[9] A. V. García and S. Fosse-Parisis. FreePhone audio tool. http://www-sop.inria.fr/rodeo/fphone. High-Speed Networking Group, INRIA Sophia Antipolis.

[10] O. A. Hellal, E. Altman, A. Jean-Marie, and I. A. Kurkova. On loss probabilities in presence of redundant packets and several traffic sources. *Performance Evaluation*, 36-37:486–518, 1999.

[11] L. Kleinrock. *Queuing systems*. John Wiley, New York, 1976.

[12] I. Kouvelas, O. Hodson, V. Hardman, and J. Crowcroft. Redundancy control in real-time Internet audio conferencing. In *Proceedings of Audio–Visual Services over Packet Networks (AVSPN)*, Aberdeen, Scotland, UK, Sept. 1997.

[13] C. Perkins and O. Hodson. Options for repair streaming media. *Request for Comments 2354*, June 1998.

[14] C. Perkins, O. Hodson, and V. Hardman. A survey of packet loss recovery for streaming audio. *IEEE Network*, 12(5):40–48, Sept. 1998.

[15] M. Podolsky. *Transmission of Real-time Multimedia Over the Internet.* PhD thesis, University of California, Berkeley, 1999.

[16] M. Podolsky, C. Romer, and S. McCanne. Simulation of FEC-based error control for packet audio on the Internet. In *Proceedings of the IEEE Infocom*, volume 2, pages 505–515, San Francisco, California, USA, Mar. 1998.

[17] T. M. Project. RAT: Robust Audio Tool. Multimedia Integrated Conferencing for European Researchers, University College London.

[18] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne. Adaptive playout mechanisms for packetized audio applications in wide-area networks. In *Proceedings of the IEEE Infocom*, June 1994.

[19] J. Rosenberg, L. Qiu, and H. Schulrinne. Integrating packet FEC into adaptive voice playout buffer algorithms on the Internet. In *Proceedings of the IEEE Infocom*, Mar. 2000.

[20] M. R. Salamatian. *Transmission Multimédia Fiable Sur Internet.* PhD thesis, Université Paris Sud UFR Scientifique d'Orsay, Dec. 1999.

[21] J. Scourias. Overview of the global system for mobile communications. Univ. of Waterloo.

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

[23] L. Takács. *Combinatorial Methods in the Theory of Stochastic Processes.* John Wiley and Sons, 1967.

[24] T. E. Tremain. The government standard linear predictive coding algorithm: LPC-10. *Speech Technology*, 1:40–49, Apr. 1982.