

Adaptive Joint Playout Buffer and FEC Adjustment for Internet Telephony

Catherine Boutremans

LCA - I & C

School of Computer and

Communication Sciences

EPFL

CH-1015 Lausanne, Switzerland

Email : Catherine.Boutremans@epfl.ch

Jean-Yves Le Boudec

LCA - I & C

School of Computer and

Communication Sciences

EPFL

CH-1015 Lausanne, Switzerland

Email : Jean-Yves.LeBoudec@epfl.ch

Abstract— We develop a joint playout buffer and Forward Error Correction (FEC) adjustment scheme for Internet Telephony, which incorporates the impact of end-to-end delay on the perceived audio quality. We show that it provides better quality than the adjustment schemes for playout buffer and FEC that were previously published. This is important because of a threshold effect when the end-to-end delay of interactive audio is around 150 ms. We represent the perceived audio quality as a function of both the end-to-end delay and the distortion of the voice signal. We develop a joint rate/error/playout delay control algorithm that optimizes this measure of quality and is TCP-Friendly. It uses a channel model for both loss and delay. We validate our approach by simulation and show that (1) our scheme allows a source to increase its utility by avoiding an increase of the playout delay when it is not really necessary, (2) it performs better than direct combinations of existing algorithms in the cases where end-to-end delay is important and (3) adaptive delay aware FEC adjustment brings significant improvements only if it is coupled with an adaptive playout adjustment.

I. INTRODUCTION

Transport of real-time, interactive audio over IP networks often suffers from packet loss and delay variation, which calls for two types of corrective actions. First, Forward Error Correction (FEC) is used [1] to mitigate the impact of packet losses; it also increases the end-to-end delay, since the destination has to wait for the redundant packet(s) to be received in order to repair packet losses. Moreover, FEC increases the bit rate requirement of an audio source. Second, adaptive playout buffer algorithms at the receiver compensate for jitter, also at the expense of usually some additional end-to-end delay.

Bolot et al. [2] proposed an adaptive rate/error control that optimizes a subjective measure of quality and incorporates a rate control. Their algorithm supposes that the destination plays the best received copy of a given packet. They neither consider losses due to playout buffer overflow nor try to optimize the overall end-to-end delay. In particular, they do not manage the additional delay due to FEC. However, it is recognized that the end-to-end delay has a great impact on the perceived quality of interactive conversations, with a threshold effect around 150 ms [3], [4] for strongly interactive conversations, whereas many voice coders can tolerate some small loss without severe penalty. As a result, the FEC scheme

in [2] may in some cases increase the delay when it would be more important to keep delay low while accepting some small loss. An adaptive *delay aware* error control was proposed in [5] to overcome this problem. The delay aware algorithm in [5] is based on the assumptions that (1) the destination plays the best copy received and that (2) the playout delay is equal to the delay that would be used if FEC were absent, plus an additional delay to be able to use FEC (as in *rat* and *freephone* [6], [7]). Thus [5] applies the philosophy that if the source went to the trouble of adding some redundancy (FEC) then the destination should wait for the redundant information to arrive.

This philosophy is challenged by Rosenberg et al. [8], who tackle the problem of the delay introduced by FEC, in the case of non-delay aware FEC. They point out that waiting for all the redundant information is inappropriate when network loss rate is low and they propose a number of playout adaptation algorithms that provide a coupling between FEC and playout buffer adaptation. These algorithms suppose that a *play first* strategy is used at the destination, specifically, that the destination plays the first copy correctly received of a given packet. These coupled playout algorithms improve the delay performance for non-delay aware FEC schemes, since they allow a destination to not wait for the FEC packets that are not needed. However, this is less clear in the case of delay aware FEC, since the source sends FEC packets only when necessary. This is one of the questions addressed in this paper.

More generally, we consider the joint problem of FEC and playout adjustment at the source and destination of an interactive audio source, while accounting for the impact of delay in the perceived utility. We propose and analyze two solution methods of increasing complexity that solve the joint problem. The first one (“partial” method, called *N1* in this paper) adjusts the playout buffer at the destination using the results in [8]; the FEC scheme (at the source) is aware of the playout delay computed at the destination and adjusts the redundancy as a function of network characteristics and of playout delay, so as to maximize the perceived audio quality. Method *N1* supposes that the destination uses a *play first* strategy (to be consistent with the use of a coupled playout

algorithm). A second, more elaborate method (“complete” method, called *N2*), jointly chooses both the playout delay and the FEC scheme so as to maximize the perceived audio quality. In the latter method, the playout delay and the FEC scheme are parameters of the optimization problem, whereas in the former, the playout delay is adapted separately and the best FEC scheme is chosen given this playout delay. In method *N2*, we consider the use of both play first and play best strategies at the destination.

As a basis for comparison, we also use two straightforward combinations of existing FEC and playout adjustment schemes. The first (method *O1*) uses the delay aware FEC adjustment method proposed in [5] together with the classical playout adaptation, plus enough delay to wait for all FEC to be received at destination. The second (method *O2*) combines the non-delay aware FEC scheme proposed in [2] with the playout delay adjustment in [8].

In addition to the development of methods *N1* and *N2*, we address the following questions:

- 1) Are there significant quality improvements in using the complete joint method *N2*? More generally, how do the different methods compare?
- 2) Is it worth using a joint FEC/playout adaptation scheme when delay aware FEC is used or, as mentioned earlier, can we rely on the source to send only those FEC packets that are necessary and wait for all FEC to be received?
- 3) With the complete method *N2*, is it preferable to adopt a *play first* or a *play best* strategy?

Our technical approach for designing *N1* and *N2* is as follows. We start by considering the perceived audio quality as a function of the audio reconstructed rate at the destination, the packet loss rate *and* the end-to-end delay. We model the channel by a (1) packet loss process that we assume to be a Gilbert loss process and (2) a stationary delay process (not necessarily i.i.d.). We assume that the network delivers packets in sequence, a reasonable assumption since audio applications have a relatively small rate. We show that, given this assumption, our method needs only to know the marginal distribution of delays (i.e. the latter hypothesis encompasses all the needed information on the joint delay distribution). We further suppose that (3) delay and losses are stochastically independent, an assumption that makes sense if packet losses are due to active queue management schemes such as RED. Then we write our FEC and playout control problem as an optimization problem and solve it numerically. When used over the standard IP best effort service, an audio source should also control its rate in order to react to network congestion and share bandwidth, in some sense, fairly [9], [10]. This is why we also incorporate a TCP-friendly module into our design.

We designed and implemented methods *N1*, *N2*, *O1* and *O2* and found by simulation and analytical results that:

- 1) Our complete scheme *N2* performs better than the partial scheme *N1* and the combinations of existing schemes *O1* and *O2*, in the cases where end-to-end delay is

important.

- 2) Contrary to some intuition, it is worth using a joint playout adjustment algorithm when a delay aware FEC scheme is used. Classical playout delay adjustments as used in *rat* and *freephone* lead to excessive playout delay.
- 3) When used with the joint adaptation scheme *N2*, the *play first* and *play best* strategies perform similarly.

The paper is organized as follows. Section II gives the necessary background material on error control for audio applications, playout adjustment algorithms, TCP friendly rate control and audio quality measures. Section III describes our utility functions, which are adapted from the E-model. Section IV presents our main results, specifically, the derivation of our joint rate/error/playout delay control methods *N1* and *N2*. Results of simulations implemented in ns2 [11] are given in Section V.

II. BACKGROUND MATERIAL

A. Error Recovery

An audio transport protocol may cope with packet losses by [1]: (1) retransmitting dropped packets, (2) using error-concealment algorithms to correct the losses, or (3) applying Forward Error Correction (FEC) to reconstruct the missing packet.

Retransmission algorithms based on Automatic Repeat Request (ARQ) have been successful in protocols like TCP, but they are typically not acceptable for real-time audio applications since they dramatically increase the end-to-end delay. FEC, on the other hand, is an attractive alternative to ARQ since it provides relatively low-delay performance. The principle of FEC is to send redundant information, along with the original information, so that lost data can be recovered, at least partially, from this redundant information. When FEC fails to recover from a loss, applications can resort to *error concealment* algorithms at the receiver to correct the effect of missing packets [12]. Error concealment algorithms [12], [13], [14], [15] use repair mechanisms like insertion, interpolation or regeneration. These techniques work well for relatively small loss rates ($\leq 10\%$) and for small packets (4-40 ms of audio) but break down when the loss length approaches the length of a phoneme (5-100 ms). Hence, error concealment schemes should be regarded as complementary to, and not substitute for, FEC.

FEC techniques can be classified as *media independent* and *media specific*. Media independent FEC uses block codes (e.g. based on Reed-Solomon [16] or on parity codes [17], [18]) to provide redundant information. Each code takes a codeword of k data packets and generate $n - k$ additional check packets for the transmission of n packets over the network. Such a code, denoted as an (n, k) code, is able to recover all losses in the same block if and only if at least k out of n packets are received correctly. These techniques have the advantage of recovering the lost packet in a bit-exact way. On the other hand, they have the disadvantage of introducing additional delays that can be significantly large (up to $n - 1$ packet intervals).

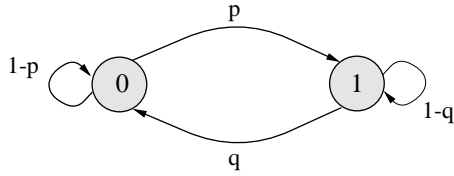


Fig. 1. The Gilbert model

Media specific (also called *signal processing*) FEC is used by most audio conferencing tools. The principle of the *signal processing* FEC is to transmit each segment of audio, encoded with different quality coders, in multiple packets. When a packet is lost, another packet containing the same segment (maybe encoded differently) can be able to cover the loss. This approach has been advocated by Hardman *et al* [12] and Bolot *et al* [19] for use on the Mbone, and extensively simulated by Podolsky *et al*. [9].

The first transmitted copy of the audio segment is called primary encoding and subsequent transmissions secondary encodings. With *signal processing* FEC, redundant audio segments are piggy-backed onto a later packet, which is preferable to the transmission of additional packets, as this decreases the amount of packet overhead and routing decisions. For example, in the case of a single redundant segment, packet n contains, in addition to its encoded samples, a redundant version of packet $n-1$. This redundant information is usually obtained using a lower-bit-rate, lower-quality encoding than the primary information. This simple scheme only recovers from isolated losses but can be modified (as proposed in [10]) to recover from consecutive losses as well by carrying in packet n redundant versions of packets $n-1$ and $n-2$, or of packets $n-1$, $n-2$ and $n-3$ or of packets $n-1$ and $n-3$ etc.

B. Loss Process of Audio Packets

The efficiency of the FEC depends on the characteristics of the loss process of audio packets. Typically, FEC is more efficient when the consecutive number of lost packets is small.

There has been many research efforts in the measurement and modeling of end-to-end Internet characteristics [20],[21],[22]. The main result is that the correlation structure of the loss process of audio packets can be modeled with low order Markov chains. In particular, a two-state Gilbert model was found to be an accurate model in many studies. Moreover, it was found that the distribution of the number of lost packets in a loss period is approximately geometric [19],[22]. These results confirmed that FEC schemes are well suited for interactive audio applications in the Internet.

The Gilbert model (depicted in Figure 1) is a two-state model in which state 1 represents a packet loss and state 0 represents a packet reaching the destination. The parameters p and q denote respectively the probabilities of passing from state 0 (no loss) to state 1 (loss) and from state 1 to state 0. In absence of redundant information, one can easily derive the packet loss rate $PLR = \frac{p}{p+q}$. This model also allows to

compute the packet loss rates after reconstruction when FEC is used. The n -stage transition matrix $P^n = [p_{ij}^{(n)}]$, $i, j \in \{0, 1\}$ is given by:

$$P^n = \frac{1}{p+q} \begin{bmatrix} q & p \\ q & p \end{bmatrix} + \frac{(1-p-q)^n}{p+q} \begin{bmatrix} p & -p \\ -q & q \end{bmatrix}$$

C. Playout Adjustment Algorithms

Jitter is compensated for at the receiver by means of adaptive playout buffer algorithms at the receiver. Most existing playout adaptation algorithms work by taking some measurements on the delays experienced by packets and updating the playout delay on a talkspurt to talkspurt basis. The main purpose of playout adaptation algorithms is to trade delay for loss. Let D be the playout delay which is defined as the difference between playout time and generation time for all the packets in a talkspurt; clearly, the larger D , the smaller the fraction of late packets.

Classical methods for playout adaptation (in absence of FEC) were proposed in [23], [24]. [23] proposed four adaptation algorithms. Each algorithm computes in some way an estimate of the mean \hat{d} and the standard \hat{v} deviation of the delays experienced by previous packets and adjusts the playout delay at the beginning of each talkspurt to be $D = \hat{d} + 4\hat{v}$. All four algorithms compute \hat{v} in the same fashion, using an exponentially weighted moving average. The four algorithms differ only in their computation of \hat{d} : Algorithms 1 and 2 in [23] compute an exponential moving average of the delays, but with different weighting factors. Algorithm 3 sets \hat{d} to be the minimum delay experienced during the prior talkspurt. Algorithm 4 performs delay spike detection. During a spike, the delay estimate tracks the delays closely, and after a spike, an exponential weighted moving average is used. For a detailed description of this algorithm, see [23]. The adaptive playout adjustment algorithm proposed in [24] tracks the network delay of recently received packets and maintains delay percentile information. This algorithm also performs spike detection. During spike mode, the delay of the first packet in a talkspurt is used as playout delay. During normal mode, the playout delay is the q th percentile among the last w packets received by the receiver.

When FEC is used, existing audio tools compute the playout delay as if FEC were absent (using one of the methods described above), compute some delay needed to make use of FEC and combine the two. If FEC is not adaptive, it is clear that this method introduces too much delay when network losses are rare. To cope with this problem, new playout adjustment algorithms were proposed in [8] to integrate more smoothly the FEC packets into the playout buffer. The algorithms proposed in [8] include:

- *Virtual delay* algorithms that are all modifications of algorithms proposed in [23], [24]. The virtual algorithms used any existing playout adaptation algorithms which compute the playout delay D as a function of the packet delays but substitute the packet delays by the packet *virtual delays* that are defined as the difference in time between the earlier of the arrival and the recovery times, and the generation time of the

packets.

- *A previous optimal* algorithm that uses the optimal delay for the previous talkspurt as playout delay for the next talkspurt. The optimal playout delay can be chosen to meet an arbitrarily chosen criteria. In [8], the optimal playout delay for a talkspurt is defined as the minimum delay that achieves a specified loss rate after reconstruction.

- *An analytical* algorithm that works by attempting to model the impact of network loss and delays on the application playout probability and the end-to-end delay. It then uses this model to find the playout delay D that meets a particular loss rate after reconstruction. For a detailed description of these coupled algorithms, see [8].

D. Rate Control

It has been suggested that audio applications share resources fairly with each other and with current TCP-based applications. One way to ensure this is to implement some form of congestion control that adapts the transmission rate in a way that *fairly* shares congested bandwidth with TCP applications, thus falling into the category of *TCP-friendly* congestion control [25].

There has been significant previous research on TCP-Friendly control mechanisms and many control schemes were proposed in the literature. Prominent examples of these schemes are: (1) window-based control mechanisms such as TEAR [26], (2) mechanisms based on additive increase, multiplicative decrease (AIMD) such as RAP [27] and LDA [28] and (3) equation-based mechanisms such as TFRC [29], and mechanisms designed specifically for video transport [30], [31] and for audio transport [5].

The rate control scheme that we use in this paper is the one described in [5]. This TCP-Friendly rate control relies on the Real-Time Transport Protocol (RTP) [32] and its control part, RTCP. With this scheme, the source performs equation-based [33] congestion control based on feedback information contained in RTCP reports and adjusts its sending rate by changing the packet size, the time interval between packets remaining constant.

E. An Audio Quality Measure: the E-Model

In this section, we give a brief overview of the E-model. A detailed description can be found in [34], [35], [36], [37].

The E-model predicts the subjective quality that is experienced by an average listener combining the impairment caused by transmission parameters (such as loss and delay) into a single rating. The rating can then be used to predict subjective user reactions, such as the Mean Opinion Score (MOS). According to ITU-T Recommendation G.107, every rating value corresponds to a speech transmission category. A rating below 60 indicates unacceptable quality, while values above 70 correspond to PSTN quality (values above 90 corresponding to very good quality).

The E-model rating R is given by:

$$R = R_0 - I_s - I_d - I_e + A \quad (1)$$

where R_0 groups the effects of noise, I_s represents impairment that occur simultaneously with the voice signal (quantization), I_d is the impairment associated with the mouth-to-ear delay, and I_e is the impairment associated with signal distortion (caused by low bit rate codecs and packet losses). The advantage factor A is the deterioration that callers are willing to tolerate because of the ‘access advantage’ that certain systems have over traditional wire-bound telephony, e.g. the advantage factor for mobile telephony is assumed to be 10. Since no agreement has been reached for the case of VoIP services, we drop the advantage factor in this study.

1) *Delay Impairment I_d* : The delay impairment I_d models the quality degradation due to the one way mouth-to-ear delay d . The delay impairment I_d is the sum of three contributing impairments:

$$I_d = I_{dte}(d, TEL) + I_{dle}(d, LEL) + I_{dd}(d) \quad (2)$$

The terms I_{dte} and I_{dle} capture the impairments due to talker and listener echo respectively. TEL and LEL are the echo losses (in dB) at the points of reflexion and depend on the echo cancellation applied. In this work, we assume that the echo losses at both ends are equal. In the sequel, we denote the echo loss by EL , with $EL = TEL = LEL$. For packetized voice calls [34], $EL = \infty$ corresponds to a perfect echo control and $EL = 51$ corresponds to a simple yet efficient echo controller. The third delay-relative impairment, I_{dd} is the loss of interactivity. In practice, the mouth-to-ear delay d is composed of the Encoding/Decoding delay (algorithmic and packetization delay), the network delay (transmission, propagation and queuing delay) and the de-jitter delay (delay spent in the playout buffer in order to cope with delay variations). Although the voice encoding and decoding process can take a significant amount of time when performed by software based implementations, the algorithmic delay is reducible by using fast hardware implementations. Without access to specific implementation information and processing delay measurement, we consider in this work that this delay is very small and can be neglected. As a consequence, the Encoding/Decoding delay is reduced to the packetization delay, that we assumed to be fixed. The playout delay as defined in Section II-C actually represents the sum of network and de-jitter delays. The mouth-to-ear delay is therefore the sum of the packetization delay and the playout delay. [35], [36] give analytical formulae to compute I_d as a function of the echo loss and the mouth-to-ear delay.

2) *Equipment Impairment I_e* : The impairments introduced by distortion are brought together in I_e . Currently, no analytical expression allows to compute I_e as a function of parameters such as the encoding rate or the packet loss rate. Estimates for I_e must be obtained through subjective measurements. A few values for I_e are given in Appendix A of [37] for several codecs and several packet loss conditions and a recent study [38] provides additional results about the impact of losses on the perceived audio quality. We build our utility function based on the values given in [37], as explained in Section III.

III. CHOICE OF UTILITY FUNCTIONS WHICH ACCOUNT FOR DELAY

We use the E-model as a basis for defining our utility functions. However, we need to further elaborate on it, for two reasons. First, our optimization framework requires an analytic expression for the rating R as a function of the encoding rate r and the packet loss rate plr , which the E-model does not readily provide. Second, we wish to define different utility functions corresponding to different modes of interactivity [39] and thus need to consider additional expressions for the delay impairment.

A. Influence of Distortion

The equipment impairment I_e captures the distortion of the original voice signal due to 1) the use of low bitrate codec and 2) packet losses (that occur in the network and in the playback buffer). In this paper, we consider these two causes of distortion separately, our goal being to approximate I_e with an expression of the form:

$$I_e(r, plr) = I_{ec}(r) + I_{el}(plr) \quad (3)$$

where I_{ec} would represent the impairment due the audio encoding and I_{el} would be the impairment due packet losses. Since we know that the effect of packet loss is to increase the measured distortion, if the distortion increases in the same way for all the codecs as a function of the packet loss rate, then this approximation would be acceptable. Let us first consider the distortion introduced by the encoder (in absence of packet loss). Various values for the intrinsic impairment of a variety of codecs can be found in [37] and [34]. Figure 2(a) shows the rating R for voice calls using different bit rate codecs, for zero end-to-end delay and no packet loss. Since these values are obtained through subjective tests, we have at our disposal only a few samples of $I_{ec}(r)$ measured for discrete values of r (let us call them $r_i, i = 1, \dots, n$) corresponding to the encoding rate of n existing codecs. Knowing the value of our utility function for discrete values of r is sufficient in the case where we wish to choose the best codec among existing ones in order to maximize the utility function of our audio source (see Section IV). However, it would be interesting to solve the problem also in the general case where the audio source would be able to encode audio samples with an infinite (large) number of rates ranging from 0 to 64 Kbits/s. This could be made possible with some sophisticated coding techniques such as multiple description coding [40]. To obtain a continuous utility function with respect to the encoding rate, we use the values given for existing codecs and simply interpolate between these values to obtain a piece-wise linear utility function (as shown in Figure 2(a)).

Now let us consider the impact of packet loss on the measured distortion. In Appendix I of [37], values of I_e are given for several codec types as a function of packet loss rate and packet loss burstiness. In this paper, we focus on results corresponding to random losses because 1) they are equivalent to the results obtained with bursty losses if the packet loss rate is small ($\leq 5\%$), which will typically be the case of the

packet loss rate after reconstruction, and 2) because a precise description of the algorithm used for generating bursty packet losses is missing from [37]. Furthermore, we assume that all codec implement some form of error concealment, which is a common practice today i.e. it is built-in in G.729A and G.723.1 and can be added on top of G.711.

In Figure 2(b), we plotted the values of $I_e - I_{ec}$ as a function of the packet loss rate plr for a variety of codecs. We can see that I_{el} increases with the packet loss rate. Moreover, we can see that the all codec follow more or less the same trajectory. Therefore, we can use a curve fitting technique to represent with a good approximation the impairment due to packet loss as a continuous function of the loss probability, independently of the codec in use. We use a logarithmic curve (see Figure 2(b)), as advised in [41] to approximate $I_{el}(plr)$ and obtain:

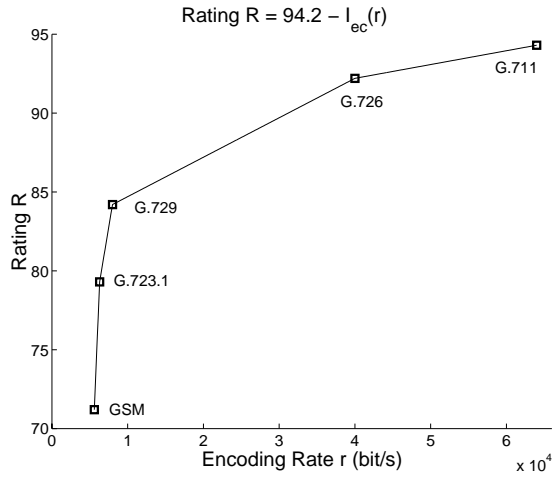
$$I_{el}(plr) = 34.3 \ln(1 + 12.8 plr) \quad (4)$$

B. Influence of End-to-End Delay

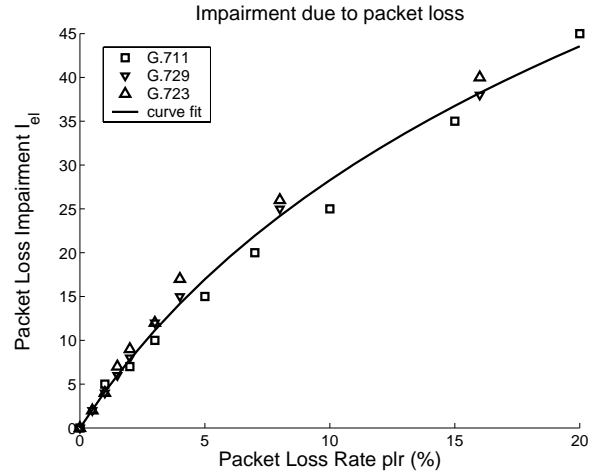
As explained in [39], there is a dimension that is not captured by the E-model, that of different modes of conversation or interactivity requirements. Different types of conversation require different switching speed and thus have different sensibilities to delay [42]. For example, a business call might require a higher level of interactivity, with shorter messages and higher switching speed than a social call. In [39], they conjecture that the fact that the E-model does not account for different modes of conversation may imply that the curves provided capture some kind of averaging of these different modes.

Besides, various studies [43], [42], [3], [4] concluded that, for natural hearing the end-to-end delay should be approximately 150 ms. While delays lower than 100 ms can not really be appreciated, delays above 150 ms are noticed by the users and become a hindrance to interactivity. As a result, the impairment caused by too long mouth-to-ear delays should be negligible for delays smaller than 150 ms and than increase rapidly as the delay gets larger than this threshold. Moreover, it is also recognized that telephony users find delays of greater than about 300 ms more like half duplex connection than a conversation [42]. In order to account for the great impact of the interactivity threshold (of 150 ms) on the quality of conversations, a utility function with a steep decrease around 150 ms was proposed in [5].

In this paper, we consider three different delay impairment functions, representing different sensibilities to the end-to-end delay. The first delay impairment function $I_{d1}(d)$ considered (see Figure 3) is based on the utility function proposed in [5] and characterizes a user with strong interactivity requirements. The second and the third delay impairment functions considered are based on the E-model: $I_{d2}(d) = I_d(d, 51)$ represents a user that is annoyed by delay because of echo, but without a clear threshold effect and $I_{d3}(d) = I_d(d, \infty)$ represents a user that attaches a small importance to delay. Figure 3 shows the influence of the mouth-to-ear delay on the utility function for the three different levels of interactivity.



(a) Rating as a function of the encoding rate (for delay=0 and no loss)



(b) Impairment due to loss as a function of the packet loss rate.

Fig. 2. Influence of distortion on quality.

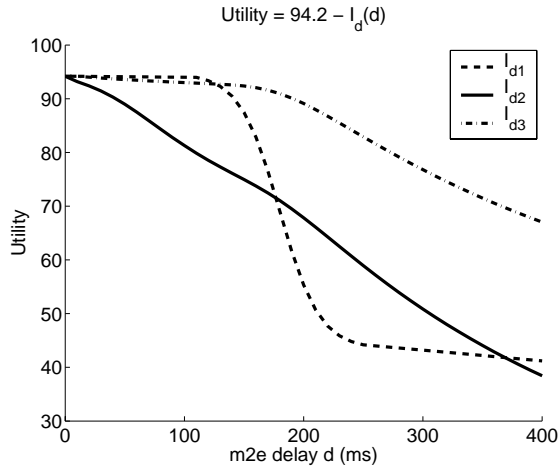


Fig. 3. Utility as a function of the mouth-to-ear delay for different interactivity levels for $r = 64K\text{bits/s}$ and no loss.

C. Our Utility Functions

In summary, we consider three different utility functions $f_i : \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow [0, 100]$, $i = 1, \dots, 3$ that correspond to three different interactivity requirements:

$$f_i(r, d, plr) = 94.2 - I_{di}(d) - I_{ec}(r) - I_{el}(plr) \quad (5)$$

where d is the one-way mouth-to-ear delay, r is the codec bit rate and plr is the residual packet loss rate (after FEC is used), $I_{di}(d)$ ($i = 1, \dots, 3$), $I_{ec}(r)$ and $I_{el}(plr)$ are defined above.

IV. ADAPTIVE JOINT PLAYOUT DELAY AND FEC ADJUSTMENT

In this section, we present two methods that solve the joint problem of FEC and playout adjustment at source and

destination, while accounting for the impact of delay in the perceived utility. The “partial” method N1 adjusts the playout buffer at the receiver using the results in [8]; the FEC scheme at the source is aware of the playout delay computed at destination and adjust redundancy so as to maximize the perceived utility. If the source uses Signal Processing FEC (SP FEC) with method N1, a play first strategy has to be used (since it is the strategy used with coupled playout algorithms). If the source uses media independant FEC like Reed-Solomon FEC (RS FEC), since all the copies are identical, there is only one possible strategy (since the first copy is also the best). The “complete” method, on the other hand, jointly chooses both the playout delay and the redundancy at the source so as to maximize the perceived utility. With method N2, no constraint is imposed on the decoding strategy. We thus consider both the play first and play best strategy when signal processing FEC is used. In summary, we study five possible combinations of FEC and playout adjustment method, FEC scheme and decoding strategy: (N1, SP FEC, play first), (N2, SP FEC, play first), (N2, SP FEC, play best), (N1, RS FEC) and (N2, RS FEC).

A. General Method

Consider a voice source with the flexibility to encode its samples at a rate x such that $x \in [0, X_{max}]$ (in the general case) or $x \in R$ with $R = \{r_i, i = 1, \dots, n\}$ (if the source has a limited set of coders at her disposal).

The quality of the voice call is characterized by a utility function $f : \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow [0, 100]$ as described in Equation (5). Our utility function has now three parameters: (1) the reconstructed rate at the destination and (2) the end-to-end delay and (3) the packet loss rate. It should be pointed out that the packet loss rate considered here is the residual

packet loss rate after reconstruction of the lost packets using FEC.

The source transmits voice packets to a destination over an unreliable network about which we made some assumptions. For the control scheme, we modeled the network by:

- **a packet loss process:** Y_i which we suppose to be a Gilbert process where $Y_i \in \{0, 1\}$ (see Section II-B). If the i th packet reaches the destination, then $Y_i = 0$, otherwise, $Y_i = 1$. The parameters p and q of the Gilbert model are estimated on-line at the receiver using the maximum likelihood estimator.
- **a stationary delay process:** D_i . We suppose that the network delays of voice packets are identically distributed and follow a given distribution $F_D(d) = P(D_i \leq d)$ ¹. Moreover, we assume that the network delivers packets *in sequence*. The latter assumption can be expressed as follows:

$$Pr(D_{n+1} \leq D_n - T) = 0 \quad (6)$$

where D_n is the network delay of the n th packet and T is the time interval between two consecutive voice packets². We show that this hypothesis allows us to consider only the marginal distribution of delays and not the joint distributions.

- **independence loss-delay:** we assume that packet losses and network delays are mutually independent.

It is clear that this model is quite simple but we have to find a tradeoff between complexity and tractability since our control scheme has to be implemented in audio sources.

The “no reordering” assumption allows to obtain all needed probabilities from the sole marginal distribution F_D . For example, the joint probability $Pr(\{D_n \leq D\} \cap \{D_{n+1} > D - T\})$ is here equal to $F_D(D) - F_D(D - T)$ (see [44]).

Let R_{max} be the rate available for the audio flow. R_{max} is the result of our TCP-Friendly rate control scheme (which is introduced in Section II-D and described in details in [5]) and is updated upon reception of an RTCP receiver report.

Let D be the playout delay for each talkspurt.

Then, our general problem can be stated as follows: Given that we can send at most K_{max} copies of each voice packet, find the optimal combination of coding scheme (optimal number of copies to send and optimal encoding rate for each copy) and playout delay D so as to maximize the quality of the voice call subject to the rate constraint.

This problem can be formulated as an optimization problem and the formulation depends on the combination of (1) the FEC/playout adjustment method, (2) the error recovery technique and (3) the decoding strategy. In the following, we study the different scenarios separately.

¹The parameters p , q and the parameters characterizing the delay distribution (which are all estimated on-line at the receiver) are sent back to the source via the application specific part (APP) of the RTCP receiver reports.

² T is the packetization delay.

B. Detailed Formulation of the Sub-Problems

Due to space limitations, we do not detail the procedure followed to obtain the mathematical expressions presented in this section but all the details can be found in [44].

1) *Method N1 - SP FEC - play first:* In this case, the optimization problem can be expressed as follows (P1):

$$\begin{aligned} \text{Given:} & \quad p, q, R_{max}, F_D(d) \text{ and } D \\ \text{maximize} & \quad \sum_{i \in \Gamma} P_{play}(i) f(x_i, D + T, PLR_{FEC}) \\ \text{over all} & \quad \Gamma, (x_i, i \in \Gamma) \\ \text{subject to} & \quad \begin{cases} \sum_{i \in \Gamma} x_i + R_{overhead} \leq R_{max} \\ x_i \geq r_0, i \in \Gamma \end{cases} \end{aligned}$$

where x_i is the encoding rate of the copy placed in i th position in the stream; T is the packetization delay³; $P_{play}(i)$ is the probability that the i th copy is sent to the audio driver; $R_{overhead}$ is the bandwidth overhead of the IP/UDP/RTP headers and PLR_{FEC} is the packet loss rate after reconstruction. Γ is the set of copies of a given packet that are sent by the source i.e., $\Gamma = \{i | \text{copy placed in } i\text{th position in the stream was sent}\}$. A value $i = 1$ corresponds to the primary information, thus the minimal $\Gamma = \{1\}$ corresponds to a source that sends no redundant information. Without loss of generality, we assume that the set Γ is an ordered set, i.e. if $\Gamma(k)$ denotes the k th element in Γ , we can write $\Gamma(k) < \Gamma(k+1)$. $P_{play}(i)$ is the probability that the i th copy is the first copy correctly received and that it is on time.

$P_{play}(i)$ is a function of the playout delay D , the parameters of the Gilbert model p and q and of the redundancy scheme Γ and is given by:

$$P_{play}(i) = F_D(D - (i-1)T) \times a_i \quad (7)$$

with $a_i =$

$$\begin{cases} \pi_0 & \text{if } i = 1 \\ \pi_1 p_{10}^{(i-1)} & \text{if } i > 1 \text{ and } I_d(i) = 2 \\ \pi_1 \prod_{j=2}^{I_d(i)-1} p_{11}^{(\Gamma(j)-\Gamma(j-1))} p_{10}^{(i-\Gamma(I_d(i)-1))} & \text{if } i > 1 \text{ and } I_d(i) > 2 \end{cases}$$

where $I_d(i)$ is the index of the i th copy in the ordered set Γ (e.g. if $\Gamma = \{1, 3\}$, $I_d(3) = 2$). As expected, we can see that $P_{play}(i)$ is an increasing function of D .

The residual packet loss rate after reconstruction PLR_{FEC} is given by:

$$PLR_{FEC} = 1 - \sum_{i \in \Gamma} P_{play}(i) \quad (8)$$

The objective function above represents the average quality measured at the destination. In the discrete case, the formulation remains the same but the second constraint is replaced by $x_i \in R, i \in \Gamma$.

³Since the playout delay is defined as the difference between playout time and generation time of for all the packets in a talkspurt, the mouth-to-ear delay is the sum of playout delay and packetization delay.

2) *Method N2 - SP FEC - play first*: In this case, the objective function is the same as in (P1) but the input parameters have changed. The problem (P2) is thus expressed as follows:

$$\begin{aligned} \text{Given:} & \quad p, q, R_{max}, F_D(d) \\ \text{maximize} & \quad \sum_{i \in \Gamma} P_{play}(i) f(x_i, D + T, PLR_{FEC}) \\ \text{over all} & \quad \Gamma, (x_i, i \in \Gamma), D \\ \text{subject to} & \quad \begin{cases} \sum_{i \in \Gamma} x_i + R_{overhead} \leq R_{max} \\ x_i \geq r_0, i \in \Gamma \end{cases} \end{aligned}$$

3) *Method N2 - SP FEC - play best*: The problem (P3) is expressed as follows:

$$\begin{aligned} \text{Given:} & \quad p, q, R_{max}, F_D(d) \\ \text{maximize} & \quad \sum_{\Delta \subseteq \Gamma, \Delta \neq \emptyset} P(\Delta) \max_{i \in \Delta} f(x_i, D + T, PLR_{FEC}) \\ \text{over all} & \quad \Gamma, (x_i, i \in \Gamma), D \\ \text{subject to} & \quad \begin{cases} \sum_{i \in \Gamma} x_i + R_{overhead} \leq R_{max} \\ x_i \geq r_0, i \in \Gamma \end{cases} \end{aligned}$$

where x_i is the encoding rate of the copy placed in i th position in the stream, $R_{overhead}$ is the bandwidth overhead of the IP/UDP/RTP headers and PLR_{FEC} is the packet loss rate after reconstruction.

The random variable $\Delta = \{i | \{Y_i = 0\} \cap \{D_i \leq D - (i - 1)T\}, i \in \Gamma\}$ represents the set of copies of a given packet that are received at the destination before the playout time of this packet. Without loss of generality, we assume that the set Δ is an ordered set, i.e. if $\Delta(k)$ denotes the k th element in Δ , we can write $\Delta(k) < \Delta(k + 1)$.

$P(\Delta)$ is the probability to receive *exactly* the set Δ before playout time and is given by:

$$\begin{aligned} P(\Delta) &= \sum_{j=K_\Delta}^{K-1} P_j(\Delta) \times \\ & \quad (F_D(D - (j - 1)T) - F_D(D - jT)) \\ & \quad + P_K(\Delta) F_D(D - (K - 1)T) \end{aligned} \quad (9)$$

where K is the position of the last copy of a given packet sent by the source, K_Δ is the position of the last copy sent by the source and received on time, i.e. $K_\Delta = \Delta(n_\Delta)$ where n_Δ is the number of elements in Δ and $P_j(\Delta)$ is given by:

$$P_j(\Delta) = \begin{cases} \pi_{b_1} & \text{if } j = 1 \\ \pi_{b_1} \prod_{k=1}^{nc_j - 1} p_{b_k b_{k+1}}^{(\Gamma(k+1) - \Gamma(k))} & \text{if } j > 1 \end{cases}$$

where nc_j is the number of elements in Γ that are inferior or equal to j and b_k is a binary value defined as $b_k = 1 - I(\Gamma(k) \in \Delta)$.

PLR_{FEC} is the probability that none of the copies arrives on time, and is given by:

$$PLR_{FEC} = 1 - \sum_{\substack{\Delta \in \Gamma \\ \Delta \neq \emptyset}} P(\Delta) \quad (10)$$

4) *Method N1 - RS FEC*: The problem (P4) is expressed as follows:

$$\begin{aligned} \text{Given:} & \quad p, q, R_{max}, F_D(d), D \\ \text{maximize} & \quad f(x, D + T, PLR_{FEC}) \\ \text{over all} & \quad (n, k), x \\ \text{subject to} & \quad \begin{cases} x \frac{n}{k} + R_{overhead} \leq R_{max} \\ x \geq r_0 \\ k \leq n \end{cases} \end{aligned}$$

where x is the encoding rate of each copy, $R_{overhead}$ is the bandwidth overhead of the IP/UDP/RTP headers and PLR_{FEC} is the packet loss rate after reconstruction and is given by:

$$PLR_{FEC} = \frac{1}{k} \sum_{i=1}^k \left\{ \frac{q}{p+q} (1 - F_D(D)) + \frac{p}{p+q} (1 - P_{REC}(i)) \right\} \quad (11)$$

where

$$P_{REC}(i) = \sum_{g=k+1}^{n-1} P_{PAR}(k, g, i) \times [F_D(D - (g - i)T) - F_D(D - (g + 1 - i)T)] + F_D(D - (n - i)T) P_{PAR}(k, n, i)$$

$$P_{PAR}(k, g, i) = \sum_{l=1}^{g-k} \sum_{m=0}^{\min(l-1, i-1)} R(m+1, i) R(l-m, g-i+1)$$

with

$$R(m, n) = \begin{cases} P_R(n) & \text{for } m = 1 \text{ and } n \geq 1 \\ \sum_{i=1}^{n-m+1} p_r(i) R(m-1, n-i) & \text{for } 2 \leq m \leq n \end{cases}$$

$$P_R(i) = \begin{cases} 1 & \text{if } i = 1 \\ q(1-p)^{i-2} & \text{otherwise} \end{cases}$$

and

$$p_r(i) = \begin{cases} 1 - q & \text{if } i = 1 \\ q(1-p)^{i-2} p & \text{otherwise} \end{cases}$$

5) *Method N2 - RS FEC*: The problem (P5) is the same as (P4) except that D is no longer an input but a parameter. (P5) is thus expressed as follows:

$$\begin{aligned} \text{Given:} & \quad p, q, R_{max}, F_D(d) \\ \text{maximize} & \quad f(x, D + T, PLR_{FEC}) \\ \text{over all} & \quad (n, k), x, D \\ \text{subject to} & \quad \begin{cases} x \frac{n}{k} + R_{overhead} \leq R_{max} \\ x \geq r_0 \\ k \leq n \end{cases} \end{aligned}$$

C. Resolution

We implemented numerical methods to solve the different optimization problems formulated above. We used a combination of the algorithm proposed in [45] (which is well suited to maximize a sum of weighted utility functions like ours and gives exact solutions) and of an exhaustive search on discrete parameters (like Γ, n, k etc.). We have a running implementation of our audio source/destination in ns2. The method is optimal if the channel complies with the model described above.

V. SIMULATION RESULTS

We investigated the behaviour of the different FEC/playout adjustments schemes under a wide range of loss and delay conditions. The results presented here constitute a small sample of our simulation results but are representative of the behaviours we could observe in our simulations. For more details about simulation results, we invite the reader to refer to [44].

We consider a simple scenario where n audio sources share a bottleneck link with $3n$ Sack TCP and an ON/OFF source (CBR 500 Kbits/s when ON) with ON and OFF periods exponentially distributed, with average ON and OFF times of 3s. Packet loss rate is varied artificially by changing the number of TCP and audio connections sharing the link.

Figure 4 shows the performance (in terms -from left to right- of mouth-to-ear delay (playout delay + 20 ms of packetization delay), utility measured at the receiver, residual packet loss rate and rate of the reconstructed audio stream at the receiver) for the different methods N1, N2, O1 and O2 as a function of the number of connections sharing the link. For all the graphs in the figure, the bottleneck bandwidth is 5Mbits/s and the propagation delay is 70 ms. The graphs show the mean values averaged over the last 300 s of simulation and over all audio connections. Figures 4 (a) to (d), (e) to (h) and (i) to (l) correspond respectively to the results obtained with utility functions f_1 (high interactivity requirements), f_2 (delay is a concern but without threshold effect) and f_3 (delay is marginally important) with a Signal Processing FEC coding scheme. Figures 4 (m) to (p) were obtained with utility function f_1 and a Reed-Solomon error coding. Algorithm 1 in [23] was used to adjust the playout delay with method O1 and its virtual version was used with N1 and O2. We tested all the playout adjustment algorithms proposed [8] but do not show the results here. The conclusions presented in the sequel remain the same for other existing playout adjustment algorithms.

Figures 4 (a) to (p) show that:

- method N2 always gets the higher utility compared to other methods. The gain in utility is significant when utility f_1 is used (Fig. 4 (b)) with a mouth-to-ear delay 20 to 30 ms smaller than with other methods (Fig. 4 (a)); the gain in utility is smaller with utilities f_2 (Fig. 4 (f)) and f_3 (Fig. 4 (j)) but still visible, with a mouth-to-ear delay 10 to 20 ms smaller than with other methods. This shows that N2 succeeds in trading delay for losses

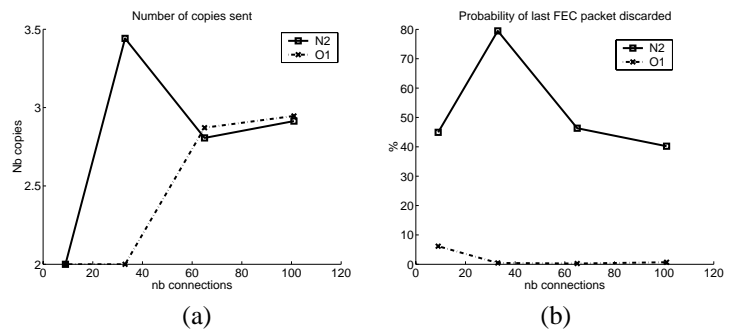


Fig. 5. The optimal solution (N2) does not always wait for all FEC packets to arrive.

while keeping the residual packet loss rate acceptable. If delay is important (Fig. 4 (a)), N2 manages to keep the mouth-to-ear delay much lower than other methods (20 to 30 ms lower) at the price of a slightly higher residual packet loss rate (see Figure 4 (c)). On the other hand, if delay issues are less crucial for the user, it accepts to increase its playout in order to reduce the residual packet loss rate (Fig. 4 (i) and (k)).

- method N1 gives results equivalent to the best combination of existing methods. When method O2 is used with Signal Processing FEC, the optimal parameter setting is $K = 3$ (K being the total number of copies sent) for a wide range of network conditions; with Reed-Solomon (n, k) , a paradoxical result is that the setting $k = 1$ give very good result. With this setting, Reed-Solomon is equivalent to a Signal Processing FEC with all the copies encoded with the same quality.

Figures 4 (q) to (t) compare the performance of the play first and play best strategies when used with N2 and Signal Processing FEC. In all our simulations, we observed that the two strategies lead to similar results. Consequently, we recommend the use of the play first strategy, which is more simple.

Figure 5 (a) shows the average number of copies of a given packet that was sent by the source when using the methods N2 and O1 with utility f_2 . Figure 5 (b) shows the probability that the last copy of given packet is discarded at destination because it arrived too late. Figure 5 (b) shows that the optimal solution does not always wait for all the FEC packets to arrive (in 40 to 50% of the cases, method N2 does not wait for the last FEC packet to be received). The playout adjustment scheme used by O1 thus leads to excessive playout delay (as can be seen in Figure 4 (a), (e) and (i)).

VI. CONCLUSIONS

We designed a method for the joint control of delay-aware FEC and playout for interactive audio applications over the Internet. We have shown that, in cases where delay matters (i.e. around a threshold effect), there is a real benefit in using the joint method. We have also shown that the improvement

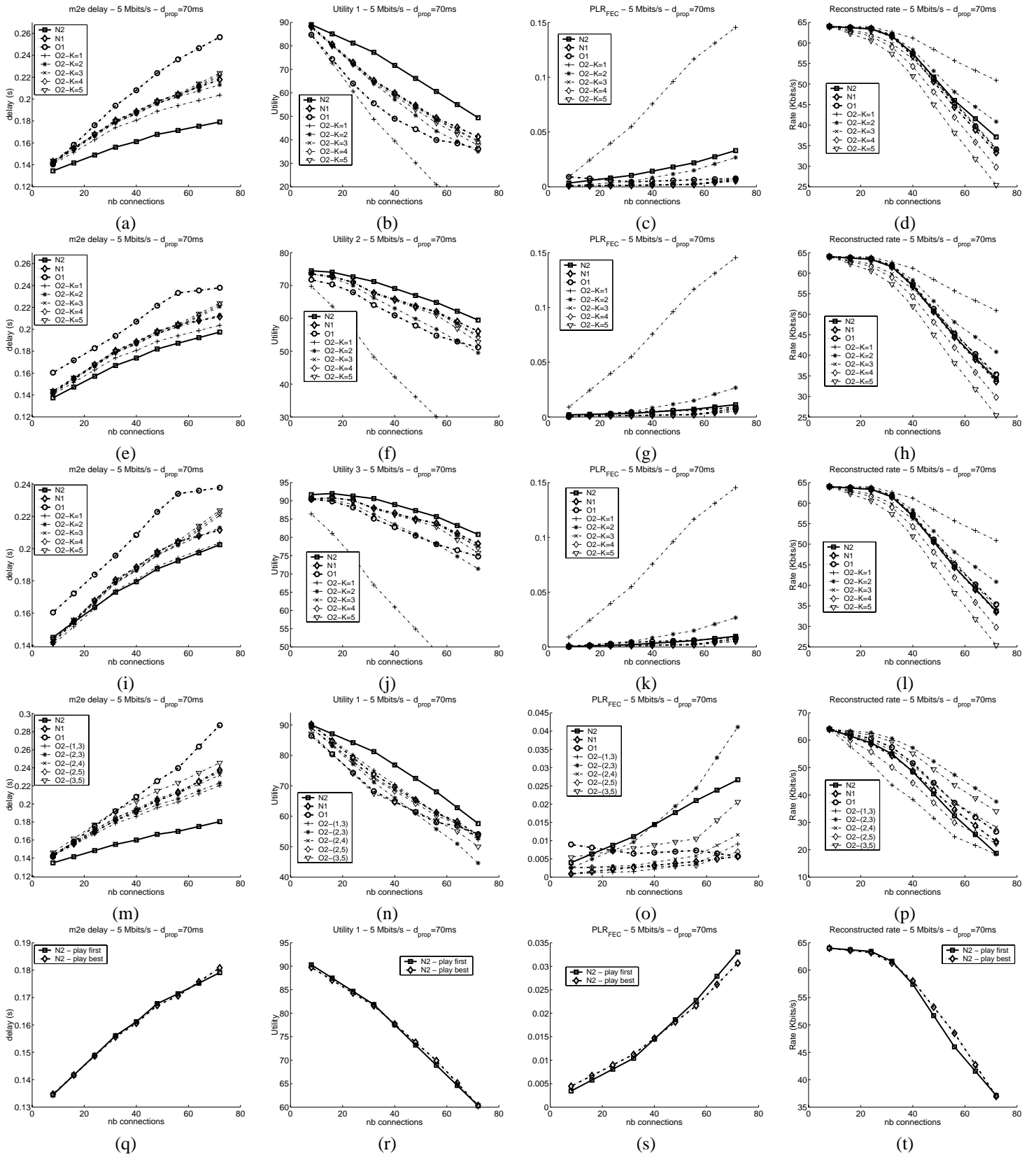


Fig. 4. Performance of the different methods $N1$, $N2$, $O1$ and $O2$ (with different FEC parameter settings). Mouth-to-ear delay, utility measured at the destination, residual packet loss rate and reconstructed rate of audio for SP FEC with utility f_1 from (a) to (d), f_2 from (e) to (h), f_3 from (i) to (l) and for Reed-Solomon with utility f_2 from (m) to (p). Comparison of play first and play best strategies from (q) to (t).

brought by delay aware FEC cannot be obtained if the delay aware FEC control is simply piggybacked onto existing adaptive playout control methods; in contrast, it should be incorporated in a complete joint optimization of both FEC and playout. We have implemented our method in ns2 and made it available for public use. Our control method uses a channel model for both delay and loss that is fairly simplistic. In further work, we will address whether there is any benefit in using more sophisticated models.

REFERENCES

- [1] C. S. Perkins, O. Hodson, and V. Hardman, "A Survey of Packet-Loss Recovery Techniques for Streaming Audio," September/October 1998.
- [2] J.-C. Bolot, S. F. Parisi, and D. Towsley, "Adaptive FEC-based error control for Internet Telephony," in *Infocom'99*, March 1999.
- [3] "One-Way Transmission Time," ITU-T Recommendation G.114, February 1996.
- [4] "Control of Talker Echo," ITU-T Recommendation G.131, August 1996.
- [5] C. Boutremans and J.-Y. Le Boudec, "Adaptive Delay Aware Error Control for Internet Telephony," in *2nd IP-Telephony workshop*, April 2001, pp. 81–92, Columbia University, New York.
- [6] "Robust Audio Tool," <http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/>.
- [7] "Freephone," <http://www-sop.inria.fr/rodeo/index.html>.
- [8] J. Rosenberg, L. Qiu, and H. Schulzrinne, "Integrating packet FEC into adaptive voice playout buffer algorithms on the internet," in *Proc. IEEE Infocom'2000*, March 2000.
- [9] M. Podolsky, C. Romer, and S. McCanne, "Simulation of FEC-based Error Control for packet audio in the Internet," in *IEEE Infocom'98*, April 1998.
- [10] J.-C. Bolot and A. V. Garcia, "Control Mechanisms for packet audio in the Internet," in *IEEE Infocom'96*, March 1996.
- [11] "Network Simulator," Available from <http://www-mash.cs.berkeley.edu/ns>.
- [12] V. Hardman, A. Sasse, M. Handley, and A. Watson, "Reliable Audio for Use over the Internet," in *Proceedings of INET'95*, June 1995.
- [13] H. Sanneck, "Concealment of lost speech packets using adaptive packetization," in *Proceedings IEEE Multimedia Systems 1998*, Austin, TX, June 1998.
- [14] "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)," March 1996.
- [15] H. Sanneck and N. Le, "Speech property-based FEC for Internet Telephony applications," in *Proceedings of the SPIE/ACM SIGMM Multimedia Computing and Networking Conference (MMCN)*, San Jose, CA, January 2000, pp. 38–51, <ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann0001:Speech-FEC.ps.gz>.
- [16] R. Blahut, *Theory and Practice of Error control codes*. Addison-Wesley, 1993.
- [17] N. Shacham and P. M. Kenney, "Packet recovery in high-speed networks using coding and buffer management," in *Proc. IEEE Infocom'1990*, vol. 1, June 1990, pp. 124–131.
- [18] D. R. Figueiredo and E. de Souza e Silva, "Efficient mechanisms for recovering voice packets in the Internet," in *Proc. IEEE Globecom'99*, 1999, pp. 1830–1836.
- [19] J.-C. Bolot and A. V. Garcia, "The case for FEC-based error control for packet audio in the Internet," in *to appear in ACM Multimedia Systems*.
- [20] V. Paxson, "End-to-end internet packet dynamics," in *Proc. ACM Sigcomm'97*, september 1997, Cannes, France.
- [21] J.-C. Bolot, "End-to-end packet delay and loss behavior in the internet," in *Proc. ACM Sigcomm'93*, August 1993, pp. 189–199, San Francisco, CA.
- [22] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modelling of temporal dependence in packet loss," in *Proc. IEEE Infocom'99*, March 1999, New York, NY.
- [23] 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*, 1994.
- [24] S. B. Moon, J. Kurose, and D. Towsley, "Packet Audio Playout Delay Adjustment: Performance Bounds and Algorithms," *ACM/Springer Multimedia Systems*, vol. 5, pp. 17–28, January 1998.
- [25] J. Mahdavi and S. Floyd, "TCP-Friendly unicast rate-based flow control," in *Draft posted on end2end mailing list*, January 1997, http://www.psc.edu/networking/papers/tcp_friendly.html.
- [26] I. Rhee, V. Ozdemir, and Y. Yi, "TEAR: TCP emulation at receivers - flow control for multimedia streaming," Department of Computer Science, NCSU, Technical report, April 2000.
- [27] R. Rejaie, M. Handley, and D. Estrin, "RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the internet," in *Proc. IEEE Infocom'99*, March 1999, New York.
- [28] D. Sisalem and H. Schulzrinne, "The loss-delay adjustment algorithm: A TCP-Friendly adaptation scheme," in *NOSSDAV'98*, July 1998, Cambridge, UK.
- [29] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *SIGCOMM'2000*, August 2000.
- [30] W. Tan and A. Zakhor, "Error resilient packet video for the internet," in *ICIP'98*, vol. 3, October 1998, pp. 458–462.
- [31] J.-C. Bolot and T. Turletti, "Experience with rate control mechanisms for packet video in the Internet," *ACM Computer Communication Review*, vol. 28, no. 21, January 1998.
- [32] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," RFC 1889, 1996.
- [33] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proc. SIGCOMM'98*, 1998.
- [34] D. de Vleeschauwer, J. Janssen, G. H. Petit, and F. Poppe, "Quality bounds for packetized voice transport," *Alcatel Telecom Review*, pp. 19–24, January 2000.
- [35] "The E-Model, a computational model for use in transmission planning," ITU-T Recommendation G.107, May 2000.
- [36] "Speech processing, Transmission and Quality aspects (STQ); Overall Transmission Plan Aspects for Telephony in Private Networks," ETSI Guide 201 050, February 1999.
- [37] "Provisional Planning values for Equipment Impairment Factor I_e ," Appendix to ITU-T Recommendation G.113, February 2001.
- [38] W. Jiang and H. Schulzrinne, "Comparison and Optimization of Packet Loss Repair Methods on VoIP Perceived Quality under Bursty Loss," in *NOSSDAV'02*, May 2002, pp. 73–81.
- [39] A. Markopoulou, F. Tobagi, and M. Karam, "Assessment of VoIP quality over internet backbones," in *to appear in Proc. Infocom'02*, June 2002.
- [40] V. A. Vaishampayan, "Design of multiple description scalar quantizers," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 821–834, May 1993.
- [41] R. Cole and J. Rosenbluth, "Voice over IP Performance Monitoring," *Computer Communication Review*, vol. 31, no. 2, pp. 9–24, April 2001.
- [42] N. Kiatawaki and K. Itoh, "Pure Delay Effect on Speech Quality in Telecommunications," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 4, pp. 586–593, May 1991.
- [43] T. J. Kostas, M. S. Borella, I. Sidhu, G. M. Schuster, J. Grabiec, and J. Mahler, "Real-Time Voice over Packet-Switched Networks," *IEEE Network*, pp. 18–27, January/February 1998.
- [44] C. Boutremans and J.-Y. Le Boudec, "Adaptive Joint Playout Buffer and FEC Adjustment for Internet Telephony," EPFL-I&C-ICA, <http://icwww.epfl.ch/publications/>, Tech. Rep. IC/2002/35, May 2002.
- [45] R. Rajkumar, C. Lee, J. Lehoczy, and D. Siewiorek, "A Resource Allocation Model for QoS Management," in *IEEE Real-Time Systems Symposium*, December 1997.