Laboratoire
d'Informatique
de Robotique
et de Microélectronique
de Montpellier
LIRMM

Rapport de Recherche

No : 03-003

Avril 2003

On the Efficiency of Forward Error Correction at the Packet Level

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

CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier

161 rue Ada, 34392 Montpellier cedex 5

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

On the Efficiency of Forward Error Correction at the Packet Level

Yvan Calas Alain Jean-Marie

Avril 2003

Abstract

It is well known that packet losses in networks are mainly due to congestion because of the finite buffer capacity of queues in routers. Forward Error Correction (FEC) can recover from losses without retransmissions of lost packets. However, FEC increases the aggregate load. In this paper, we investigate the tradeoff between correction of losses and the increase in the individual packet loss rate. Oguz et al. have proposed to use a Reed-Solomon erasure code. This coding technique makes a distinction between packets bearing relevant information (data packets) and packets of redundancy. They have studied the global packet loss rate which does not make this distinction. We argue that since redundancy packets do not contain the original relevant information, it is only by measuring the loss of data packets that an exact evaluation of the loss rate is obtained. Accordingly, we extend and refine the analytical model of Oguz et al. For different configurations of the system, we present results for several loss metrics and the goodput. The results show that if most of the sources are not implementing FEC then FEC offers a better goodput. Otherwise, if all sources are FEC sources, the loss rate increases considerably making FEC inefficient.

Keywords: FEC, loss metrics, goodput, Markov chain.

Résumé

Il est couramment admis que les pertes de paquets dans les réseaux sont essentiellement causées par la capacité finie des buffers des files d'attente situées au niveau des routeurs. FEC (*Forward Error Correction*) peut réparer les pertes de paquets sans avoir recours à la retransmission des paquets perdus. Cependant, FEC augmente la charge agrégée du réseau. Ce papier étudie le compromis entre la correction des pertes et l'augmentation du taux de perte des paquets. Oguz et Ayanoglu ont proposé l'utilisation d'un code correcteur d'erreur Reed-Solomon. Ce codage fait la distinction entre les paquets portant l'information utile (paquets d'information) et les paquets de redondance. Ils ont étudié le taux de perte de paquet global sans faire cette distinction. Comme les paquets de redondance ne contiennent pas d'information utile, nous montrons que c'est uniquement en mesurant le taux de perte des paquets d'information qu'une évaluation exacte du taux de perte est faite. Pour cela, nous étendons et affinons le modèle analytique d'Oguz et Ayanoglu. Pour différentes configurations du système, nous présentons les résultats pour plusieurs métriques de perte et pour le débit utile. Les résultats montrent que si la plupart des sources n'utilisent pas FEC, alors FEC offre un meilleur débit utile. Cependant, si toutes les sources sont FEC, le taux de perte augmente considérablement et FEC devient alors inefficace.

Mots-clés : FEC, métriques de perte, débit utile, chaîne de Markov.

1 Introduction

This paper deals with the problem of packet loss in networks like ATM networks or the Internet. It is well known that losses are mainly due to congestion because of the finite buffer capacity of queues in routers (buffer overflow) [11]. Indeed, losses at the link level are extremely low : for instance optical fibers offer an insignificant bit error rate (BER) of about 10^{-14} [15].

There exists two alternative error control techniques that can decrease losses. ARQ (Automatic Repeat Request), which is a closed-loop technique, is based on the use of error or loss detecting codes and on packet retransmissions by the source upon destination's request. ARQ requires cooperation between the source and the destination. One of the drawbacks of this technique is that, because of the retransmissions, ARQ is not appropriate for audio and/or video applications requiring strong time constraints : the retransmission time of packets is at least equal to a round trip time.

The second technique called FEC (Forward Error Correction) is an open-loop technique. It consists in adding to packets a redundancy which is exploited by the destination to recover from losses without requiring packet retransmission. This extra information is computed using error correcting codes. FEC is well suited for real-time applications which can tolerate few losses. However, FEC has two antagonistic effects : on the one hand the redundancy generated by the source increases the overall load of the network resulting in an increase of the loss rate; on the other hand the redundant information generated by FEC helps to recover from a part of the losses. Therefore, FEC utilization is only interesting if the loss rate increase can be compensated by a greater correction of losses.

The analysis of this compromise is the topic of several recent research papers in the area of networking : Shacham [14] analyzed single-parity code under the independent packet loss assumption. Later, Shacham and McKenney [15] extended this study and showed by simulations that the independent packet loss assumption in the analysis may yield overly optimistic results. Biersack [5, 4] carried out extensive simulations by using a code over consecutive packets generated by a packet stream derived from a compressed motion picture interfering with bursty streams. Cidon *et al.* [6] introduced recursive schemes to compute the loss probability of j packets out of a FEC block of n consecutive packets for various packet arrival models and service distributions. These recursions have been solved by Altman *et al.* [2, 3, 7] by using multi-dimensional probability generating functions for single or multiple sessions. The authors investigated also the effect of redundancy on the loss probability.

Oguz and Ayanoglu [9] studied a mixed traffic composed by a foreground traffic (traffic created by FEC sources which generate redundancy) and a background traffic (generated by non-FEC sources). For this purpose, they modeled an output-buffered ATM multiplexer with a discrete-time Markov chain for the output buffer occupancy with the traffic information explicitly incorporated. They also used a Reed-Solomon Erasure correcting code (RSE) as described in [8] to generate the redundant data. This coding technique generates two distinct types of packets : data packets also called information packets (packets bearing the relevant information) and parity packets also called redundancy packets (packets bearing the redundancy). The metric studied in [9] is the *global* packet loss rate (PLR) which does not make the distinction between data packets and parity packets.

We start from the model of Oguz and Ayanoglu with the purpose of refining and extending their analysis. First of all, we make the distinction between data packets and parity packets while computing the PLR. We show that the PLR is not the exact metric for the evaluation of losses since the parity packets are only used to recover lost data packets. Therefore we argue that the parity packets should not be taken into account for the computation of the loss rate. As parity packets do not contain the original message, it is only by measuring the losses of information packets that an exact evaluation of the loss rate is obtained. Furthermore, we propose additional metrics, such as packet loss rate before correction, packet loss rate of information before correction, message loss rate and goodput. We compare packet loss metrics under various conditions of traffic.

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

2 Analytical framework

In this section, we recall the model introduced by Oguz and Ayanoglu [9]. We briefly describe the overall topology of the system and the Markov chain used in [9]. We also recall the properties of the Reed-Solomon erasure correcting code presented in [8].

2.1 Topology of the system

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

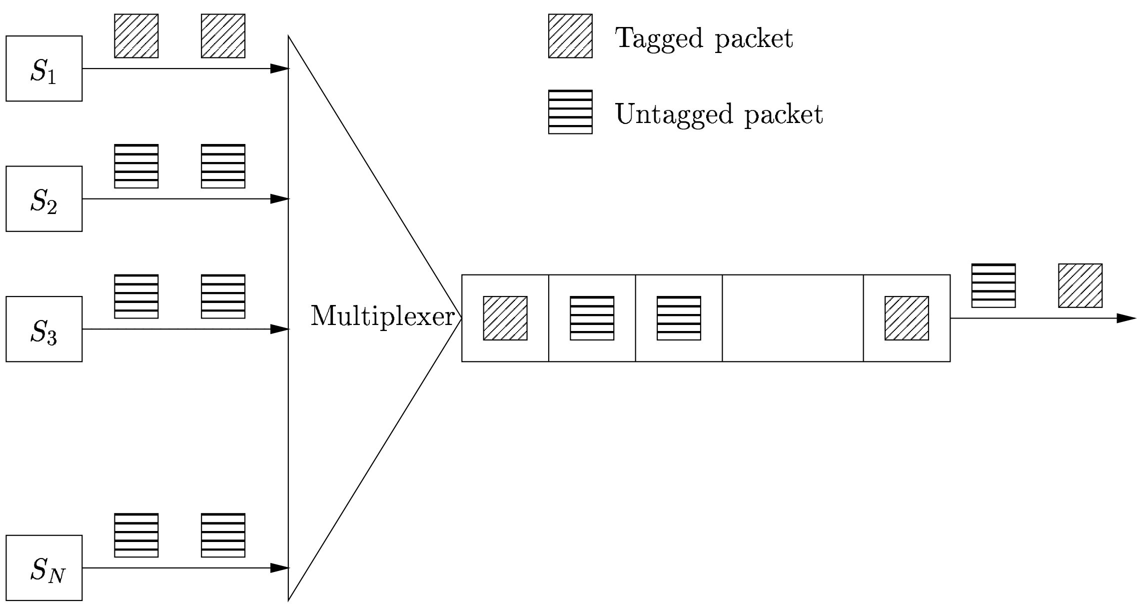


Figure 1: Topology of the system.

2.2 Source model

The source model used in this paper is a discrete-time on/off model which alternates between active state (1) and idle state (0) periods as shown in figure 2. Let $q_{1,tt'}$ be the transition probability of one source from state t to state t' with $t, t' \in \{0,1\}$, and let α and β be respectively the idle-to-idle and active-to-active state transition probabilities. At each slot, the state moves from the active to the idle state with probability $\delta = 1 - \beta$ and from the idle to the active state with probability $\gamma = 1 - \alpha$. We assume that a state transition takes place just prior to the end of a time slot and that a packet is generated at the beginning of a slot if the new state is the active state.

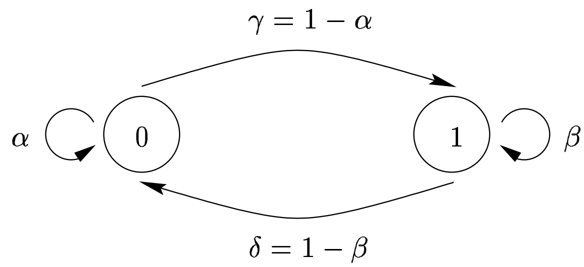


Figure 2: Discrete-time on/off source model.

By construction, the length of a burst of packets is geometrically distributed and the probability for a burst to be of length k is equal to $\beta^{k-1}(1-\beta)$. Moreover, the average time for a source to stay in the active state (respectively in the idle state) is $1/(1-\beta)$ (respectively $1/(1-\alpha)$).

Therefore, the stationary probability of the active state is the *normalized load* offered by one non-FEC source and is given by (1) :

$$\rho_1 = \frac{1-\alpha}{2-\alpha-\beta}. \tag{1}$$

Let ρ_{et} be the load offered by a FEC source. ρ_{et} increases with the number h of parity packets generated by the FEC source for each group of k packets (refer to [9] for further details), and is equal to :

$$\rho_{et} = \rho_1 \Big(1 + \frac{h}{k}\Big). \tag{2}$$

As h increases, the load offered by the FEC source becomes heavier than the load offered by one non-FEC source. This effect is obtained by increasing the active-to-active transition probability. We denote this new transition probability by β' and we compute it as follows :

$$\beta' = 2 - \alpha - \frac{1-\alpha}{\rho_{et}}. \tag{3}$$

Consequently, we denote by δ' the active-to-idle state transition probability of the FEC source which is equal to $\delta' = 1 - \beta'$.

The *normalized aggregate load*, which is defined as the total load generated by N non-FEC sources is then computed as :

$$\rho = \sum_{m=1}^{N} \rho_1 = N\rho_1,$$

since the sources are identical in the abscence of coding.

When the traffic generated by one FEC source is mixed with $N-1$ non-FEC sources, the *aggregate load* becomes :

$$\rho = \rho_{et} + \sum_{m=1}^{N-1} \rho_1 = \rho_{et} + (N-1)\rho_1.$$

Moreover, if the N sources generate parity packets, each source increases its normalized load by a factor of $1 + h/k$ (as seen in (2)) and the aggregate load will be equal to :

$$\rho = N\rho_{et} = N\rho_1\Big(1 + \frac{h}{k}\Big).$$

From the superposition of $N-1$ independent and identical on/off sources (corresponding to the $N-1$ non-FEC sources), it results an aggregate source which is modelized by a N states discrete-time Markov chain where each state represents the number of active non-FEC sources. The behavior of the aggregate arrival process is well described in the literature [10, 9]. Given that u sources were active in the previous slot, the transition probability $q_{N-1,uu'}$ that u' sources (among $N-1$ non-FEC sources) are active in the current slot is equal to :

$$q_{N-1,uu'} = \sum_{l=\underline{l}}^{\bar{l}} \binom{u}{l} \delta^l \beta^{u-l} \binom{N-1-u}{u'-u+l} \gamma^{u'-u+l} \alpha^{N-1-u'-l}, \tag{4}$$

where $\underline{l} = \max(0, u-u')$ and $\bar{l} = \min(u, N-1-u')$.

If the $N-1$ sources become FEC sources, then their load increases in the same manner as for a single FEC source and (4) becomes :

$$q_{N-1,uu'} = \sum_{l=\underline{l}}^{\bar{l}} \binom{u}{l} \delta'^l \beta'^{u-l} \binom{N-1-u}{u'-u+l} \gamma^{u'-u+l} \alpha^{N-1-u'-l}.$$

2.3 The Markov chain

The Markov chain described in [9] is modelized by a state space $\mathcal{S}$ where each state is represented by a triple (b,t,u). For a given slot, $b \in \{0, \dots, B\}$ is the number of packets contained in the queue having a buffer capacity of size B, $t \in \{0,1\}$ is the state (0 if idle, 1 if active) of the tagged (FEC) source in the slot and $u \in \{0, \dots, N-1\}$ is the number of non-tagged sources which generate a packet in the slot. Some states are considered inconsistent due to the boundary conditions at $b = 0$ or $b = B$. Therefore, $\mathcal{S}$ is defined as follows :

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

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

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

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

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

and

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

$\mathcal{S}_L$ adds new states in the Markov chain. We define by $\mathcal{S}_E$ the set of all the states of the Markov chain by :

$$\mathcal{S}_E = \mathcal{S}_D \cup \mathcal{S}_S \cup \mathcal{S}_L.$$

Let $Q = (q_{ij})_{(i,j)\in\mathcal{S}_E\times\mathcal{S}_E}$ be the transition matrix of the Markov chain. The transition probability from state $i = (b, t, u)$ to state $j = (b', t', u')$ is given by :

$$q_{ij} = q_{1,tt'} q_{N-1,uu'}, \tag{5}$$

where $b' = \min(B, \max(b + t' + u' - 1, 0))$.

Figures 3 and 4 illustrate the shape of the Markov chain.

2.4 Reed-Solomon erasure correcting code

We consider a Reed-Solomon Erasure correcting code (RSE) as described in [8] to generate the parity packets. For k data packets $d_1, d_2, \ldots, d_k$, the RSE encoder generates h parity packets $p_1, p_2, \ldots, p_h$ bearing only redundant information useful for the recovery of lost data packets without making any modification to the original data. The concatenation of the k data packets and the h parity packets is called a *FEC block* $\{d_1, d_2, \ldots, d_k, p_1, p_2, \ldots, p_h\}$ of size $n = k + h$. The k data packets will be referred to as *data block* as defined in [5, 4]. If these k data packets are transmitted without loss to the destination, it is not useful to recover the eventual lost parity packets as they do not contain relevant data. It is only in case of loss of data packets that packet recovery is required. If the sum of the loss of data and parity packets is at most h packets, the RSE decoder at the destination can retrieve successfully all lost packets. As a result all the relevant information is saved. Otherwise, if the total loss exceeds h packets, it is impossible to recover the lost packets. Usually, in this case, the packets which were correctly transmitted are considered as lost as well since the are part of an incomplete message. Nevertheless, it is useful for audio and video applications to keep the data packets that have been transmitted without loss since such applications can resort to other receiver-based mechanisms like insertion, interpolation or regeneration [12].

3 Loss rate modeling

The authors in [9] used the recursive formula (6) to compute the packet loss rate without making any distinction between data packets and parity packets. In this section, we propose a novel recursive formula (8) which makes a clear distinction between these two types of packets while computing packet loss probabilies : indeed, what is computed is the probability to lose m data packets among k data packets and l parity packets among h parity packets for a FEC block of size $n = k + h$. The algorithm associated to (8) will be presented in Section 5. This new result allows us to compute metrics that we will present in Section 4.

3.1 A recursive formula without distinction between loss of data and parity packets

Let $f_i(v,w)$ be the probability to lose v tagged packets among w tagged packets when the system is in the state i at the arrival of the first of them, and let q_{ij} be the transition probability from state i to state j in the Markov chain. Oguz and Ayanoglu, in [9], give the recursive formula (6) and an iterative algorithm to compute $f_i(v,w)$ for all $i \in \mathcal{S}_E$:

$$f_i(v,w) = \sum_{j\in\mathcal{S}_D} q_{ij} f_j(v,w) + \sum_{j\in\mathcal{S}_S} q_{ij} f_j(v,w-1) + \sum_{j\in\mathcal{S}_L} q_{ij} f_j(v-1,w-1). \tag{6}$$

It should be noted that this formula does not make distinction between losses of data packets and losses of parity packets while computing $f_i(v,w)$.

If we consider a FEC block of size n, we can deduce from $f_i(v,n)$ the probability distribution $F(v,n)$ to lose v tagged packets among n tagged packets :

$$F(v,n) = \frac{1}{\rho_{et}} \sum_{i\in\mathcal{S}_S\cup\mathcal{S}_L} p_i f_i(v,n), \tag{7}$$

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

3.2 Making the distinction between data and parity packets

We extend the computation by defining $g_i(m,k;l,h)$ as the probability to lose m tagged data packets among the next k tagged data packets and to lose l tagged parity packets among h tagged parity packets assuming that the initial state is $i \in \mathcal{S}_E$. This definition allows us to make the distinction between packets of information and packets bearing redundancy and to derive the following properties :

- Property 1 : $\forall i \in \mathcal{S}_E, g_i(0,0;0,0) = 1$ since the probability to lose 0 data packets among 0 data packets and 0 parity packets among 0 parity packets is equal to one.
- Property 2 : $g_i(m,k;l,h) = 0$ if $m \notin \{0,1,\ldots,k\}$ or $l \notin \{0,1,\ldots,h\}$.
- Property 3 : $g_i(m,k;0,0) = f_i(m,k)$, where $f_i(m,k)$ is computed using (6).
- Property 4 : $g_i(0,0;l,h) = f_i(l,h)$.

Conditioning on the first transition of the Markov chain, we obtain the recursive formula (8) where $\mathbb{1}_{\{A\}}$ is the event-indicator function which is equal to 1 if condition A inside is true and is equal to 0 otherwise. For all $i \in \mathcal{S}_E$,

$$g_i(m,k;l,h) = \sum_{j\in\mathcal{S}_D} q_{ij} g_j(m,k;l,h) + b_i, \tag{8}$$

where :

$$\begin{aligned}
b_i &= \left(\sum_{j\in\mathcal{S}_S} q_{ij}g_j(m,k-1;l,h)\right)\times \mathbb{1}_{\{0\le m<k\}}\\
&+ \left(\sum_{j\in\mathcal{S}_S} q_{ij}g_j(m,k;l,h-1)\right)\times \mathbb{1}_{\{0\le l<h, m=k=0\}}\\
&+ \left(\sum_{j\in\mathcal{S}_L} q_{ij}g_j(m-1,k-1;l,h)\right)\times \mathbb{1}_{\{0<m\le k\}}\\
&+ \left(\sum_{j\in\mathcal{S}_L} q_{ij}g_j(m,k;l-1,h-1)\right)\times \mathbb{1}_{\{0<l\le h, m=k=0\}}.
\end{aligned} \tag{9}$$

Equation (9) can be simplified using property 4 as follows :

$$\begin{aligned}
b_i &= \left(\sum_{j\in\mathcal{S}_S} q_{ij}g_j(m,k-1;l,h)\right)\times \mathbb{1}_{\{0\le m<k\}}\\
&+ \left(\sum_{j\in\mathcal{S}_S} q_{ij}f_j(l,h-1)\right)\times \mathbb{1}_{\{0\le l<h\}}\\
&+ \left(\sum_{j\in\mathcal{S}_L} q_{ij}g_j(m-1,k-1;l,h)\right)\times \mathbb{1}_{\{0<m\le k\}}\\
&+ \left(\sum_{j\in\mathcal{S}_L} q_{ij}f_j(l-1,h-1)\right)\times \mathbb{1}_{\{0<l\le h\}}.
\end{aligned}$$

The loss of a packet depends on the type of the arrival state j reached in the next slot. If $j \in \mathcal{S}_L$, the packet is lost, otherwise if $j \in \mathcal{S}_S$ it is not lost. The lost packet can be a data or parity packet. In case of a data packet we decrease the parameters m and k by one unit and in case of a parity packet we decrease the parameters l and h by one unit. If the packet is not lost and if $j \in \mathcal{S}_S$, parameter k (respectively parameter h) is decreased by one unit if the packet is a data packet (respectively if the packet is a parity packet). Finally if the FEC source did not generate any packet in the slot, then $j \in \mathcal{S}_D$. As a result, $g_j(m,k;l,h)$ is computed for the newly reached state j.

For k and h fixed and for all $i \in \mathcal{S}_E$, $m \in \{0,1,\ldots,k\}$ and $l \in \{0,1,\ldots,h\}$, once the values of $g_i(m,k,l,h)$ are computed, the probability distribution to lose m among k tagged information packets and l among h tagged redundant packets within a FEC block of size $n = k+h$ is given by (10) under the following distribution :

$$G(m,k;l,h) = \frac{1}{\rho_{et}} \sum_{i\in\mathcal{S}_S\cup\mathcal{S}_L} p_i g_i(m,k;l,h), \tag{10}$$

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

We also use this distribution to compute a set of metrics (presented in the next section) and obtain in Section 6 a set of figures which illustrates the behavior of packet loss in the presence of redundancy under various conditions of traffic.

4 Metrics

With the model presented in Section 3, it becomes possible to compute useful metrics for a better analysis of packet losses. The metrics are divided into two categories : those which compute packet losses before correction (pre-recovery metrics) and those which compute packet losses after correction (post-recovery metrics). We first present the pre-recovery metrics and continue the presentation with the post-recovery metrics.

4.1 Pre-recovery metrics

We present here three metrics related to the packet loss rate before correction : packet loss rate of information (PLRIBC), packet loss rate of redundancy (PLRRBC) and packet loss rate without distinction between data packets and parity packets (PLRBC).

Let $X_{bc} \in \{0, 1, \ldots, k\}$ and $Y_{bc} \in \{0, 1, \ldots, h\}$ be two discrete random variables which characterize the number of lost data packets (respectively the number of lost parity packets) in a FEC block before a correction performed by the RSE decoder. We note respectively by $\mathbb{E}X_{bc}$ and $\mathbb{E}Y_{bc}$ the average number of data packets (respectively parity packets) lost in a FEC block before correction.

For $m \in \{0, 1, \ldots, k\}$, the probability to lose m data packets in a FEC block is given by (11) :

$$\mathbb{P}(X_{bc} = m) = \sum_{l=0}^{h} G(m, k; l, h), \tag{11}$$

and the mean of the random variable X_{bc} is :

$$\begin{aligned} \mathbb{E}X_{bc} &= \sum_{m=0}^{k} m\mathbb{P}(X_{bc} = m) \\ &= \sum_{m=0}^{k} \sum_{l=0}^{h} mG(m, k; l, h). \end{aligned}$$

Similarly to (11), the probability to lose l parity packets in a FEC block for $l \in \{0, 1, \ldots, h\}$ is :

$$\mathbb{P}(Y_{bc} = l) = \sum_{m=0}^{k} G(m, k; l, h),$$

and the mean of the random variable Y_{bc} is given by :

$$\mathbb{E}Y_{bc} = \sum_{l=0}^{h} \sum_{m=0}^{k} lG(m, k; l, h).$$

We then have the following definitions :

Definition 1 (PLRIBC) *For k and h fixed, the packet loss rate of information before correction is defined as the ratio of the average number of information packets lost in the data block before correction to the size of the data block :*

$$\begin{aligned} PLRIBC(k, h) &= \frac{\mathbb{E}X_{bc}}{k} \\ &= \frac{1}{k} \sum_{m=0}^{k} \sum_{l=0}^{h} mG(m, k; l, h). \end{aligned}$$

Definition 2 (PLRRBC) *For k and h fixed, the packet loss rate of redundant packets before correction is defined as the number of parity packets lost in a FEC block before correction divided by the number of parity packets :*

$$\begin{aligned} PLRRBC(k,h) &= \frac{\mathbb{E}Y_{bc}}{h} \\ &= \frac{1}{h}\sum_{m=0}^{k}\sum_{l=0}^{h} lG(m,k;l,h). \end{aligned}$$

Definition 3 (PLRBC) *For k and h fixed, the packet loss rate before correction is defined as the ratio of the average number of lost packets in a FEC block before correction to the size of the FEC block :*

$$\begin{aligned} PLRBC(k,h) &= \frac{\mathbb{E}(X_{bc}+Y_{bc})}{k+h} \\ &= \frac{1}{k+h}\sum_{m=0}^{k}\sum_{l=0}^{h}(m+l)G(m,k;l,h). \end{aligned}$$

The three metrics defined above are easy to compute since all values of $g_i(m,k;l,h)$ for k and h fixed can be computed by Algorithm 1 presented in Section 5.

4.2 Post-recovery metrics

We present in this section several metrics related to packet losses after the recovery performed by the RSE decoder. We first recall how the packet loss rate (PLR) is computed in [9] when there is no distinction between data and parity packets and we show that the PLR can be deduced from our model introduced in Section 3. We then present the computation of a new metric, the packet loss rate of information (that is the loss rate of data packets). We finally show how to compute the goodput and the message loss rate (MLR).

Let X and Y be two discrete random variables which correspond respectively to the number of lost data packets (respectively parity packets) in a FEC block after recovery. We note respectively by $\mathbb{E}X$ and $\mathbb{E}Y$ the average number of lost data packets (respectively parity packets) in a FEC block after correction.

For $m \in \{0,\dots,k\}$, the probability to lose exactly m packets of information in a FEC block is equal to (12) since the number of lost packets (without distinction between data and parity packets) within a FEC block is greater than h :

$$\mathbb{P}(X=m) = \sum_{l=0}^{h} G(m,k;l,h)\times \mathbb{1}_{\{m+l>h\}}. \tag{12}$$

The mean of the random variable X is then deduced from (12) :

$$\mathbb{E}X = \sum_{m=0}^{k} m \sum_{l=0}^{h} G(m,k;l,h)\times \mathbb{1}_{\{m+l>h\}}.$$

Similarly :

$$\mathbb{P}(Y=l) = \sum_{m=0}^{k} G(m,k;l,h)\times \mathbb{1}_{\{m+l>h\}},$$

and :

$$\mathbb{E}Y = \sum_{l=0}^{h} l \sum_{m=0}^{k} G(m,k;l,h) \times \mathbb{1}_{\{m+l>h\}}.$$

4.2.1 Packet loss rate

In [9], the packet loss rate, PLR, (which is also called cell loss rate or CLR) is computed by using (6) and (7) and is equal to :

$$PLR(k,h) = \frac{1}{k+h} \sum_{v=h+1}^{k+h} vF(v,k+h).$$

With our notation, the PLR is defined as follows :

Definition 4 (PLR) *For k and h fixed, the packet loss rate after the correction performed by the decoder is defined as the ratio of the expected number of unrecoverable losses in a FEC block to the size of the FEC block. Therefore the PLR is equal to :*

$$\begin{aligned} PLR(k,h) &= \frac{\mathbb{E}(X+Y)}{k+h} \\ &= \frac{1}{k+h} \sum_{m=0}^{k} \sum_{l=0}^{h} (m+l) G(m,k;l,h) \times \mathbb{1}_{\{m+l>h\}}, \end{aligned}$$

since a FEC block is lost when more than h packets are lost from the block.

4.2.2 Packet loss rate of information

The PLR as described above is not the exact metric for the evaluation of losses since the parity packets are only used to recover lost data packets. It is only by measuring the losses of information packets that an exact evaluation of the loss rate is obtained. To compute the information packet loss rate (PLRI), we have to count only lost packets of information assuming that $G(m,k;l,h)$ is known, $m \in \{0,1,\ldots,k\}$ and $l \in \{0,1,\ldots,h\}$.

We define the PLRI as follows :

Definition 5 (PLRI) *The PLRI is defined as the ratio of the average number of lost data packets after correction, to the size of a data block :*

$$\begin{aligned} PLRI(k,h) &= \frac{\mathbb{E}X}{k} \\ &= \frac{1}{k} \sum_{m=0}^{k} \sum_{l=0}^{h} mG(m,k;l,h) \times \mathbb{1}_{\{m+l>h\}}. \end{aligned}$$

Note that for $h=0$, the PLRI is equal to the PLRIBC.

In addition to the metrics of loss rate presented above, we present in the following subsection an additional metric related to the goodput.

4.2.3 Goodput

Let ρ_u be the goodput in the system, i.e. the throughput of data packets saved by FEC after the correction. We define ρ_u as follows :

Definition 6 (Goodput) *If we consider a FEC block, ρ_u is the ratio of the number of saved data packets, to the time needed to transmit all the data packets of the FEC block.*

If we note by $\bar{k} = k - \mathbb{E}X$ the average number of data packets not repared after decoding, we have :

$$
\begin{aligned}
\rho_u &= \rho_1 \frac{\bar{k}}{k} \\
&= \rho_1 \frac{k - \mathbb{E}X}{k} \\
&= \rho_1 \left(1 - \frac{\mathbb{E}X}{k}\right) \\
&= \rho_1 \left(1 - PLRI(k,h)\right).
\end{aligned} \tag{13}
$$

It follows from (13) that the goodput is a function of the PLRI. In [9], no distinction between information and redundancy packets is made. The authors do not use goodput as a metric, but if they had, the corresponding "goodput" ρ_s would have logically been defined as the number of *saved packets* among a FEC block after the correction performed by FEC. In this case, ρ_s would have been written as :

$$
\begin{aligned}
\rho_s &= \rho_{et} \frac{n - \mathbb{E}(X+Y)}{k+h} \\
&= \rho_{et} \left(1 - PLR(k,h)\right).
\end{aligned}
$$

We present in the next subsection another loss metric that evaluates the loss of a whole message.

4.2.4 Message loss rate

A message is said to be lost if there is more than h packets lost (without distinction between data and parity packets) among the $k+h$ packets. In this case, exactly the k packets of the data block and the h parity packets are considered to be lost even if some packets of the FEC block are well arrived at the destination.

Definition 7 (Message Loss Rate) *For a FEC block, we define the MLR as the probability to lose the whole block. It is equal to $\mathbb{P}(X_{bc} + Y_{bc} > h)$, or :*

$$
MLR(k,h) = \sum_{m=0}^{k} \sum_{l=0}^{h} G(m,k;l,h) \times \mathbb{1}_{\{m+l>h\}}.
$$

From this definition, we observe that the MLR measures the rate of losses in a system where every packet of the FEC block (including those that were transmitted correctly) are considered as lost, and the data packets that arrived correctly at destination are not taken into account.

This technique is not convenient for audio and video applications as for this kind of applications it is preferable to keep the data packets arrived correctly instead of considering them as lost. Nevertheless, the MLR remains an interesting metric as it has been well studied in literature [5, 4, 6, 2] and as it can easily be computed. Moreover the MLR is a pratically useful bound of the PLR and PLRI (we will give a detailed account of this last topic in Section 6).

5 Algorithmic issues

In this section, we present an algorithm to compute $g_i(m, k; l, h)$ according to (8) and a second algorithm to compute all metrics described in the previous section. We finally conclude this section by studying the complexity of these two algorithms.

5.1 Algorithm for computing $g_i(m, k; l, h)$

Algorithm 1 is an iterative algorithm which computes all the $g_i(m, k; l, h)$ with a data block of size k and a fixed number h of parity packets. The algorithm is composed of three steps. In the first step, we initialize $g_i(0, 0; 0, 0)$ to 1 (Property 1). We then compute the LU factorization of matrix A. The value of A and the detailed explanation of this factorization will be given in Section 5.3 which deals with the complexity of the algorithm. In the second step, we compute $g_i(0, 0; l, h)$ (which are the $f_i(l, h)$ by Property 4), the probability of losing l parity packets among h parity packets, assuming we are in the state i in the Markov chain. The last step consists in computing $g_i(m, k; l, h)$ the probability of losing l data packets among k data packets, and l parity packets among h parity packets.

5.2 Algorithm for computing metrics as a function of the parameter h

Unlike in [9] where the packet loss rate for a FEC block of fixed size n is computed by varying the parameters k and h, we consider a data block of fixed size k and we vary the number h of parity packets to obtain a FEC block of variable size. The following Algorithm 2 computes all metrics presented in Section 4 for a fixed size k of the data block and the number of parity packets varying from 0 to a given parameter h_{max}.

5.3 Complexity of the algorithms

Let now consider the complexity of Algorithms 1 and 2. The computation of quantities $g_i(0, 0; l, h)$ and $g_i(v, w; l, h)$ for $i \in \mathcal{S}_D$ in steps 2 and 3 of Algorithm 1 requires the resolution of a set of $|\mathcal{S}_D|$ linear equations with $|\mathcal{S}_D|$ unknowns. These equations can be written in a matrix form $Ax = b$ where $A = (A_{ij})_{(i,j)\in\mathcal{S}_D\times\mathcal{S}_D}$ is a sparse matrix having the following coefficients :

$$A_{ij} = \begin{cases} 1 - q_{ii} & \text{if } i = j, \\ -q_{ij} & \text{if } i \neq j. \end{cases} \tag{14}$$

The vector $b = (b_i)_{i\in\mathcal{S}_D}$ is given by (9) and is formed of known quantities, already computed in previous iterations. Vector $x = (g_i(m, k; l, h))_{i\in\mathcal{S}_D}$ is the solution of the set of linear equations.

In order to reduce the complexity of the resolution of the system of linear equations, we compute first the LU factorization of matrix A (using Gaussian elimination) with complexity of at most $O(|\mathcal{S}_D|^3)$. Once the LU factorization of A is known, the resolution of

Algorithm 1 Computation of $g_i(m,k;l,h)$

Require: k,h are fixed values, matrix A.

/* First step : initialization */

$\forall i \in \mathcal{S}_E, g_i(0,0;0,0) = 1$.

Compute the LU factorization (Gaussian elimination) of matrix A.

/* Second step : computation of $g_i(0,0;l,h), \forall i \in \mathcal{S}_E, \forall l \in \{0,1,\dots,h\}$ */

if $h > 0$ **then**

  for $w = 1$ to $w = h$ **do**

    for $v = 0$ to $v = w$ **do**

      Solve $g_i(0,0;v,w)$ for all $i \in \mathcal{S}_D$ using (8) and the LU factorization of A.

    end for

    for $v = 0$ to $v = w$ **do**

      Compute $g_i(0,0;v,w)$ for all $i \in \mathcal{S}_S \cup \mathcal{S}_L$ using (8).

    end for

  end for

end if

/* Third step : computation of $g_i(m,k;l,h)$, $\forall i \in \mathcal{S}_E$, $\forall m \in \{0,1,\dots,k\}$, $\forall l \in \{0,1,\dots,h\}$ */

for $l = 0$ to $l = h$ **do**

  for $w = 1$ to $w = k$ **do**

    for $v = 0$ to w **do**

      Solve $g_i(v,w;l,h)$ for all $i \in \mathcal{S}_D$ using (8) and the LU factorization of A.

    end for

    for $v = 0$ to w **do**

      Compute $g_i(v,w;l,h)$ for all $i \in \mathcal{S}_S \cup \mathcal{S}_L$ using (8).

    end for

  end for

end for

the system of linear equations can be performed with complexity of $O(|\mathcal{S}_D|^2)$ instead of $O(|\mathcal{S}_D|^3)$.

Moreover, $g_i(0,0;v,w)$ and $g_i(v,w;l,h)$ for $i \in \mathcal{S}_S \cup \mathcal{S}_L$ can be computed directly from (8) with an insignificant complexity linear in $|\mathcal{S}_E|$. Consequently, the complexity of Algorithm 1 is $O(|\mathcal{S}_D|^3 + h^2|\mathcal{S}_D|^2 + hk^2|\mathcal{S}_D|^2)$. Based on this complexity we then deduce the complexity of Algorithm 2 which is $O\big(h_{max}(|\mathcal{S}_D|^3 + h_{max}^2|\mathcal{S}_D|^2 + h_{max}k^2|\mathcal{S}_D|^2)\big)$.

This complexity can be improved by taking into account the following two remarks :

- For $i \in \mathcal{S}_D$, the computation of $g_i(0,0;l,h+1)$ depends on the computation of $g_i(0,0;l,h)$ because we compute $g_i(0,0;v,w)$ in the second step of Algorithm 1 by varying w from 0 to $l+1$ and by varying v from 0 to w. It suffices to store the results of the computation of all $g_i(0,0;l,h)$ for $i \in \mathcal{S}_D$ and $l \in \{0,\dots,h\}$ to derive straightaway the computation of $g_i(0,0;l,h+1)$ for $i \in \mathcal{S}_D$ and $l \in \{0,\dots,h+1\}$ using (8). This way, the overall complexity of Algorithm 2 is reduced and becomes $O\left(h_{max}|\mathcal{S}_D|^3 + h_{max}^2|\mathcal{S}_D|^2(1+k^2)\right)$.
- In case of only one FEC source and $N-1$ non-FEC sources, the matrix A remains unchanged for every value of h (and therefore every value of β', see (3)), i.e. $\forall h \in \mathbb{N}, A(h) = A$. The LU factorization of A is only computed once before executing

Algorithm 2 Computation of metrics as function of the parameter h

Require: k, h_{max}.
  Compute matrix P using (4) and (5).
  Compute matrix A using (14).
  for $h = 0$ to $h = h_{max}$ **do**
    Compute $g_i(m, k; l, h)$ using Algorithm 1 for all $i \in \mathcal{S}_E$, $m \in \{0, 1, \dots, k\}$ and $l \in \{0, 1, \dots, h\}$.
    Compute $G(m, k; l, h)$ using (10) for $m \in \{0, 1, \dots, k\}$ and $l \in \{0, 1, \dots, h\}$.
    Compute metrics presented in Section 4.
  end for

Algorithm 2 and is not recomputed in Algorithm 1 for each value of h. Consequently, the complexity of Algorithm 2 is reduced to $O\left(|\mathcal{S}_D|^3 + k^2 h_{max}^2 |\mathcal{S}_D|^2\right)$. This second remark is not valid if all sources are FEC sources because the matrix A is modified according to the value of h : $\forall h, h' \in \mathbb{N}$ with $h \neq h'$, $A(h) \neq A(h')$. As a result, the LU factorization of A has to be computed for every value of h, if all sources are FEC sources.

6 Numerical results

Based on the metrics described in Section 4, we present in this section various numerical results that illustrate the packet loss process before and after the correction of losses performed by FEC. These results are obtained by varying different parameters of the system such as the buffer size B, the number of FEC and non-FEC sources, the normalized load ρ_1 of these sources and the number of parity packets h that compose a FEC block. Figure 5 to Figure 10 present these results. All these figures show the variation of the PLRIBC, the PLRI, the MLR and the goodput as a function of the number h of parity packets contained in a FEC block and also as a function of the aggregate load ρ. In all the experiments, h ranges from 0 to 31 and k is fixed to 16 packets.

The loss rate metrics (that is PLRIBC, PLRI and MLR) are represented by the y-axis located on the left hand side of the figures, whereas the goodput ρ_u is represented by the y-axis located on the right hand side of the figures. The x-axis in the upper side of the figures represents the aggregate load ρ and the x-axis in the lower side of the figures represents the number h of parity packets generated by the tagged FEC source.

At first, we study in subsection 6.1 the case where one FEC source is associated with $N-1$ non-FEC sources. We then study in subsection 6.2 the case where all sources are FEC sources. In both cases, the rate of increase of the load of every FEC source is the same : every FEC source increases its load as described in (2). This increase of the load depends on the values of the coefficients α and β that generate a traffic more or less bursty. We also analyze the MLR in subsection 6.3 and we discuss the closeness between the MLR and the PLRI; we show that the difference between them is less than one order of magnitude. Finally, in subsection 6.4, we analyze the difference between the PLR and the PLRI and show that this difference depends on traffic conditions and on the configuration of the system.

6.1 Single FEC source

Figures 5 to 7 show the efficiency of FEC for various conditions of traffic and different configurations of the system. In Figure 5, for a small buffer and a bursty traffic generated by 8 sources, even if the value of the PLRIBC is quite high (from 1.6% to 4.9%), the use of FEC allows a reduction of the information packets loss rate (a PLRI of about 9.10^{-3} for $h = 31$) of about 2.7 orders of magnitude (as compared to the PLRI measured for a traffic generated by non-FEC sources). We can also notice that more information packets are correctly transmitted when more parity packets are added in a FEC block, even though this addition causes the increase of the aggregate load.

Figure 6 shows that if the number of sources is increased to 16 and the buffer size is increased to 100 then the packet loss rate before correction (PLRIBC) is also high (from 1.4% for $h = 0$ to 6.1% for $h = 31$). Nevertheless, as the reduction of the information packets loss rate (a PLRI of about 0.08% for $h = 31$) is only about 1.9 orders of magnitude, the correction performed by FEC is less efficient than the correction carried out by FEC under the configuration of Figure 5. Unlike Figure 5, the PLRI does not decrease monotonously as a function of h. On the contrary, for $h = 1$ the PLRI reaches its maximum value and the goodput reaches its minimum value. For this configuration it is interesting to add more parity packets in the FEC block in order to increase the goodput.

Finally, in the case of Figure 7, the normalized load of every source is increased ($\rho_1 = 0.1$ in this case). We obtain a traffic more bursty than in the case of Figures 5 and 6 and a larger aggregate load of about 1.6 to 1.8. In addition, the PLRIBC as well as the PLRI are more significant : the PLRIBC is about 37.5% for $h = 0$ and about 44.2% for $h = 31$, and the PLRI decreases from 37.5% to 6%. It is interesting to notice in this case that although the aggregate load is very high, it is possible to decrease the PLRI of about 0.8 order of magnitude for $h = 31$. In addition, the goodput increases with the number of parity packets and gets closer to the normalized load ρ_1. These observations show that even if the aggregate load is high, more information packets can be transmitted correctly at destination. Likewise, in case of Figures 5 and 6, the amount of relevant information transmitted correctly increases with h.

For all these reasons, we claim that in case of a bursty traffic generated by one FEC source and $N-1$ non-FEC sources, FEC is efficient for various aggregate loads. In order to avoid as much as possible the loss of information packets in a FEC block, the FEC source should increase the size of the FEC blocks by adding as much parity packets as possible. However, the amount of redundancy is limited in practice by the delay constraints of the application (for instance audio or video), the amount of bandwidth available, and other rules like the "TCP-friendliness".

6.2 N FEC sources

If all sources use FEC, the aggregate load will increase more rapidly with the number of parity packets. The conclusions of the previous section may not hold anymore. It is then interesting to observe the behavior of the information packets loss process (before and after correction performed by FEC) for one tagged FEC source when all sources become FEC sources. Figures 8 to 10 show that FEC is not efficient in this situation. This confirms the simulation results of [5, 4].

In the case of Figure 8, we consider a small buffer ($B = 25$) and 8 FEC sources generating a bursty traffic. For this setting, we can observe that the value of PLRIBC is quite large and increases quickly with h : for $h = 0$, the PLRIBC equals 1.6% and for

$h = 31$ it is equal to 30.2%. Unfortunately, the recovery performed by FEC is not efficient since the addition of redundancy leads to an increase of the value of the PLRI. This means that more information is lost by adding redundancy than in the case of total absence of redundancy. For instance, for $h = 31$, the PLRI is equal to 3.3% which is twice the value of the PLRI (and also twice the value of the PLR) for $h = 0$. Nevertheless, for $h \geq 18$, the curve of the PLRI begins to decrease slightly, causing the increase of the goodput. This shows that above a certain threshold of h, more information packets can be transmitted correctly at destination. However, the goodput reaches too slowly the value 0.0483 for $h = 31$, which is far from the normalized load $\rho_1 = 0.5$, but seems to continue increasing for $h > 31$.

In the case of Figure 9, we consider a large buffer ($B = 100$) and 16 FEC sources generating a bursty traffic. For this setting, we can observe that the use of FEC leads again to worst results for the PLRIBC and the PLRI. Both of these metrics increase quickly and monotonously with h : for 1.4% to 57.4% for the PLRIBC and for 1.4% to 26.4% for the PLRI. This result shows, as in the case of Figure 8, that the recovery of information packets performed by FEC is inefficient since the losses are still important. But, unlike Figure 8, the goodput decreases continuously with h in the range displayed. For this configuration, the addition of redundancy gives disastrous results since it leads to a high increase of the network load without making a significant recovery. In this case FEC can not transmit effectively more information packets to the destination.

In Figure 10, we considered the same configuration as in the Figure 9, but we increased the normalized load of every source to $\rho_1 = 0.1$ so as to obtain a highly bursty traffic. This way, the aggregated load of the system is considerably increased from 1.6 to 4.7. Consequently, the curves of PLRIBC and PLRI increase monotonously with h and are almost identical. For instance, for $h = 0$ the PLRIBC is the same as the PLRI and equals 37.5%. Whereas, for $h = 31$, the PLRIBC (78.7%) is insignificantly greater than the PLRI (73.1%). This closeness between the PLRIBC and the PLRI indicates that FEC performs in this case a very low recovery from the information packet losses. The high increase in the aggregate load has also an effect on the goodput. Indeed, the goodput curve decreases and is approximatively divided by two when h increases from 0 to 31.

It should finally be noted that for such conditions of traffic (all sources being FEC sources), the information packets transmitted by a tagged source suffer from a considerable loss and hence it is particularly difficult to recover them.

We believe that in the current Internet, if most of other sources are not implementing FEC, then FEC will turn out to be profitable as it has been shown in subsection 6.1. This will not be the case, when all of the sources use FEC. In this case, the gain obtained (for the PLRI and the goodput) from the single tagged FEC source studied in subsection 6.1 is lost. Every FEC source loses more and more information packets as the number of parity packets in the FEC block is increased. In this case the use of FEC is not advised and it is preferable to only send blocks without any parity packet.

6.3 Results on the MLR

This metric is commonly used in the evaluation of losses at the message level [5, 4, 6, 2]. From Definitions 4, 5 and 7, it is clear that the MLR is an upper bound of both PLR and PLRI. Moreover, the difference between the MLR and the PLRI turns out to be less than one order of magnitude, so that the MLR may constitute an acceptable approximation of the PLRI. For instance, in the case of one FEC source, the curves of the MLR in Figures

5 to 7 tend to the curve of the PLRI for large values of h. In the same manner, the MLR remains a good approximation of the PLRI in the case of N FEC sources (as shown in Figures 8 and 10) even if this approximation is less accurate as compared to the case of one FEC source. We can also notice that the MLR curves in Figures 8 and 10 are not monotone but are almost constant. For instance, for the configuration of the Figure 10, the value of the MLR is of about 88.4% for $h = 0$, then decreases to 86.9% for $h = 5$ and finally increases to 90.7% for $h = 31$. Note that for a configuration where all sources are FEC sources, the MLR is larger than in the case of a configuration with only one FEC source. These higher values of MLR exhibit a very high loss probability of a FEC block (that is the whole message is lost with a high probability).

6.4 Distinction between PLR and PLRI

In this section, we investigate the difference between the PLR and the PLRI. For fixed k and h, we compute the absolute difference defined as follows :

$$\Delta(k,h) = PLRI(k,h) - PLR(k,h),$$

and the relative difference defined as follows :

$$\Delta_{rel}(k,h) = \frac{|\Delta(k,h)|}{PLRI(k,h)}.$$

The values of the PLR and the PLRI becomes more and more distinct when the absolute difference becomes larger. Moreover, the position of the curve of the PLRI with respect to the position of the curve of the PLR can be determined according to the sign of $\Delta(k,h)$.

For fixed k and h, if the relative difference between the PLR and the PLRI does not exceed 5% then we consider that the PLR is a reasonable approximation of the PLRI. Figure 11 (respectively Figure 12) illustrates the variation of the absolute difference (respectively the relative difference) as a function of the number of parity packets h. For both figures, the number k of information packets is fixed to 16, the buffer size is $B = 100$ and the traffic is generated by 16 bursty sources (with $\alpha = 0.995$ and $\beta = 0.905$) among which there is only one FEC source.

In the case of Figure 11, for h between 0 and 18, the PLRI curve is located above the PLR curve since $\Delta(k,h) > 0$. On the other hand, for $h > 18$, the PLRI curve is located below the PLR curve. However, for $h \geq 31$, the two curves are almost identical. Figure 12 shows a maximum relative error of about 4%. Note that the maximum absolute difference ($5.12\ 10^{-4}$ for $h = 6$) and the maximum relative difference (4.23% for $h = 7$) are not necessarily reached for the same number h of parity packets.

The maximum values of the relative difference presented by Tables 1 and 2 under various conditions of traffic and different configurations of the system are obtained for values of h between 3 and 8. The tables contain the value of the aggregate load ρ for each of these maximum. Tables 1 and 2 also show that for certain configuration of the parameters (α, β, B and N) the relative difference can be more important. We will first consider configurations with a bursty traffic ($\alpha = 0.995$, $\beta = 0.905$) and a highly bursty traffic ($\alpha = 0.99$, $\beta = 0.91$).

In case of Table 1, the traffic is generated by one FEC source and $N - 1$ non-FEC sources. In this case the absolute difference varies within the range 10^{-3} to 10^{-4} according to traffic conditions. Yet, the relative difference is above 5% for a configuration with a small buffer and 8 sources generating bursty traffic. This means in this case that the

PLR is an inaccurate approximation of the PLRI. In addition, the PLR underestimates the amount of the information packets lost. Moreover, if the burstiness of the sources is increased to $\rho_1 = 0.1$ then the maximum relative difference observed is 1.0% and the PLR becomes a reasonable approximation of the PLRI.

α	β	ρ_1	ρ	B	N	PLRI	Δ	Δ_{rel}
0.995	0.905	0.05	0.421	25	8	$9.22\ 10^{-3}$	$4.92\ 10^{-4}$	5.34%
			0.821	100	16	$1.15\ 10^{-2}$	$4.89\ 10^{-4}$	4.23%
0.99	0.91	0.1	1.643	100	16	$3.41\ 10^{-1}$	$3.65\ 10^{-3}$	1.07%
0.96	0.24	0.05	0.418	25	8	$6.01\ 10^{-15}$	$6.60\ 10^{-16}$	10.9%
			0.815	25	16	$1.91\ 10^{-6}$	$1.62\ 10^{-7}$	8.45%

Table 1: Maximum relative difference and the corresponding absolute difference between the PLR and the PLRI for the case of one FEC source and $N-1$ non-FEC sources.

In case of Table 2, the traffic is generated by N FEC sources. In this case the absolute difference is of about 10^{-3}. The relative difference is not negligible but lower than in the case of Table 1 where only one source out of the N sources is a FEC source. In this case, for a highly bursty traffic, the maximum relative difference is insignificant and is of about 0.4%.

α	β	ρ_1	ρ	B	N	PLRI	Δ	Δ_{rel}
0.995	0.905	0.05	0.55	25	8	$3.18\ 10^{-2}$	$1.40\ 10^{-3}$	4.41%
			1.1	100	16	$9.98\ 10^{-2}$	$2.86\ 10^{-3}$	2.87%
0.99	0.91	0.1	2.2	100	16	$5.25\ 10^{-1}$	$2.12\ 10^{-3}$	0.40%
0.96	0.24	0.05	0.55	25	8	$2.62\ 10^{-9}$	$2.77\ 10^{-10}$	10.5%
			0.95	25	16	$2.38\ 10^{-3}$	$9.99\ 10^{-5}$	4.19%

Table 2: Maximum relative difference and the corresponding absolute difference between the PLR and the PLRI for the case of N FEC sources.

For configurations with a smooth traffic ($\alpha = 0.96$, $\beta = 0.24$), we can observe in Tables 1 and 2 a maximum relative difference that exceeds 8%. However, the PLRI has a very low value. For instance, for $B = 25$, $N = 16$ and for a relative difference of 8.45%, we observe a PLRI of about $1.91\ 10^{-6}$. Nevertheless, if the number of sources is increased and if all sources are FEC sources, then the maximum relative difference is lower than 5%, making the PLR a reasonable approximation of the PLRI. This is illustrated by Table 2 for $\alpha = 0.96$, $\beta = 0.24$ and $N = 16$.

7 Summary and conclusions

In this paper, we have studied the performance of FEC by analyzing the tradeoff between the packet loss rate and the increase of the overall network load induced by the use of FEC. For this purpose, we used a Reed-Solomon erasure (RSE) correcting code. This coding technique plays an important role in our study because the RSE code makes the distinction between the packets bearing the relevant information and the redundant packets. However, earlier works (like [9]) did not take this fact into account in their performance analysis. For instance, the authors in [9] did not make the distinction between information and parity packets while computing the packet loss rate (PLR). Also, many previous analysis have focused on the MLR. As redundancy packets do not contain the original message, it is only by measuring the losses of information packets that an exact evaluation of the loss rate is obtained.

In the light of this observation, by extending and refining the model of [9], we have proposed a more accurate analysis based on the distinction between data and parity packets. Our extended model is based on a recursive formula which computes $g(m, k; l, h)$ the probability to lose m packets of information among k packets of information and to lose l parity packets among h parity packets. In addition, we have proposed an iterative algorithm that computes this probability and have studied its complexity. This model has allowed us to exhibit and compute several metrics that we have divided into two categories (pre-recovery metrics and post-recovery metrics). This distinction has enabled us to show clearly the contribution brought about by FEC concerning the packet loss correction. Moreover, we have shown how to compute the goodput of the FEC source.

For various conditions of traffic and different configurations of the system, results have shown that if one source out of N sources is implementing FEC, then the use of FEC is profitable. Actually, in this case, the more redundancy is added, the better the goodput. Indeed, in spite of the increase of the information packets loss rate before correction (PLRIBC) caused by the increase of the normalized aggregate load, FEC remains efficient since the information packets loss rate (PLRI) with redundancy is less than the PLRI with no redundancy. Our results have further shown that if all sources are implementing FEC then every source suffers from higher information packets loss rate as compared to the loss experimented by non-FEC sources. For this configuration, the use of FEC is not advised. In summary, we believe that if most of other sources are not implementing FEC, then FEC will turn out to be efficient. This confirms previous results obtained by simulations in [5, 4]. Similar results have also been obtained in [1] by using an analytical model based on a $M/G/1/K$ queueing system and by using a simple FEC scheme implemented in recent audio tools (Freephone [16], Rat [13]).

Finally, we have studied the absolute and relative differences between the PLR and the PLRI and we have shown that these metrics give distinct values. Based on the relative difference, we have observed that the PLR can be a reasonable or inappropriate approximation of the PLRI depending on the configuration of the system and on conditions of traffic. This is the topic of our current investigations.

References

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

[2] E. Altman and A. Jean-Marie. The Loss Process of Messages in a M/M/1/K Queue. In *Proc. IEEE INFOCOM'94*, pages 1191–1198, Toronto, Canada, June 1994.

[3] E. Altman and A. Jean-Marie. Loss Probabilities for Messages with Redundant Packets Feeding a Finite Buffer. *IEEE Journal on Selected Areas In Communications*, 16(5):778–787, June 1998.

[4] E. W. Biersack. A Simulation Study of Forward Error Correction in ATM Networks. *Computer Communication Review*, 22(1):36–47, January 1992.

[5] E. W. Biersack. Performance Evaluation of Forward Error Correction in ATM Networks. In *Proc. SIGCOMM'92*, volume 22, pages 248–257, Baltimore, Maryland, USA, August 1992.

[6] I. Cidon, A. Khamisy, and M. Sidi. Analysis of Packet Loss Processes in High-Speed Networks. *IEEE Transactions on Information Theory*, 39(1):98–108, January 1993.

[7] O. Ait Hellal, E. Altman, A. Jean-Marie, and I.A. Kurkova. On Loss Probabilities in Presence of Redundant Packets and Several Traffic Sources. *Perfomance Evaluation*, 36-37:486–518, 1999.

[8] A.J. McAuley. Reliable Broadband Communications using a Burst Erasure Correcting Code. In *Proc. ACM SIGCOMM'90*, pages 297–306, Philadelphia, PA, USA, September 1990.

[9] N.C. Oguz and E. Ayanoglu. Performance Analysis of Two-Level Forward Error Correction for Lost Cell Recovery in ATM Networks. In *INFOCOM'95 (2)*, pages 728–737, 1995.

[10] Y. Ohba, M. Murata, and H. Miyahara. Analysis of Interdeparture Processes for Bursty Traffic in ATM Networks. *IEEE Journal on Selected Areas In Communications*, 9:468–476, April 1991.

[11] H. Ohta and T. Kitami. A Cell Loss Recovery Method using FEC in ATM Networks. *IEEE Journal on Selected Areas In Communications*, pages 1471–1483, December 1991.

[12] C. Perkins, O. Hodson, and V. Hardman. A Survey of Packet-Loss Recovery Techniques for Streaming Audio. *IEEE Network*, August 1998.

[13] The Mice Project. RAT : Robust Audio Tool. http://www-mice.cs.ucl.ac.uk/multimedia/soft-ware/rat/, Multimedia Integrated Conferencing for European Researchers, University College London.

[14] N. Shacham. Packet Recovery and Error Correction in high-Speed Wide-Area Networks. In *Proc. MILCOM'89*, pages 29.5.1–29.5.7, May 1989.

[15] N. Shacham and P. McKenney. Packet Recovery in High-Speed Networks using Coding and Buffer Management. In *Proc. INFOCOM 90*, volume 1, pages 124–131, San Francisco, CA, USA, June 1990.

[16] A. Vega-Garcia and S. Fosse-Parisis. FreePhone Audio Tool. http://www-sop.inria.fr/rodeo/ fphone/, High-Speed Networking Group, INRIA Sophia Antipolis.

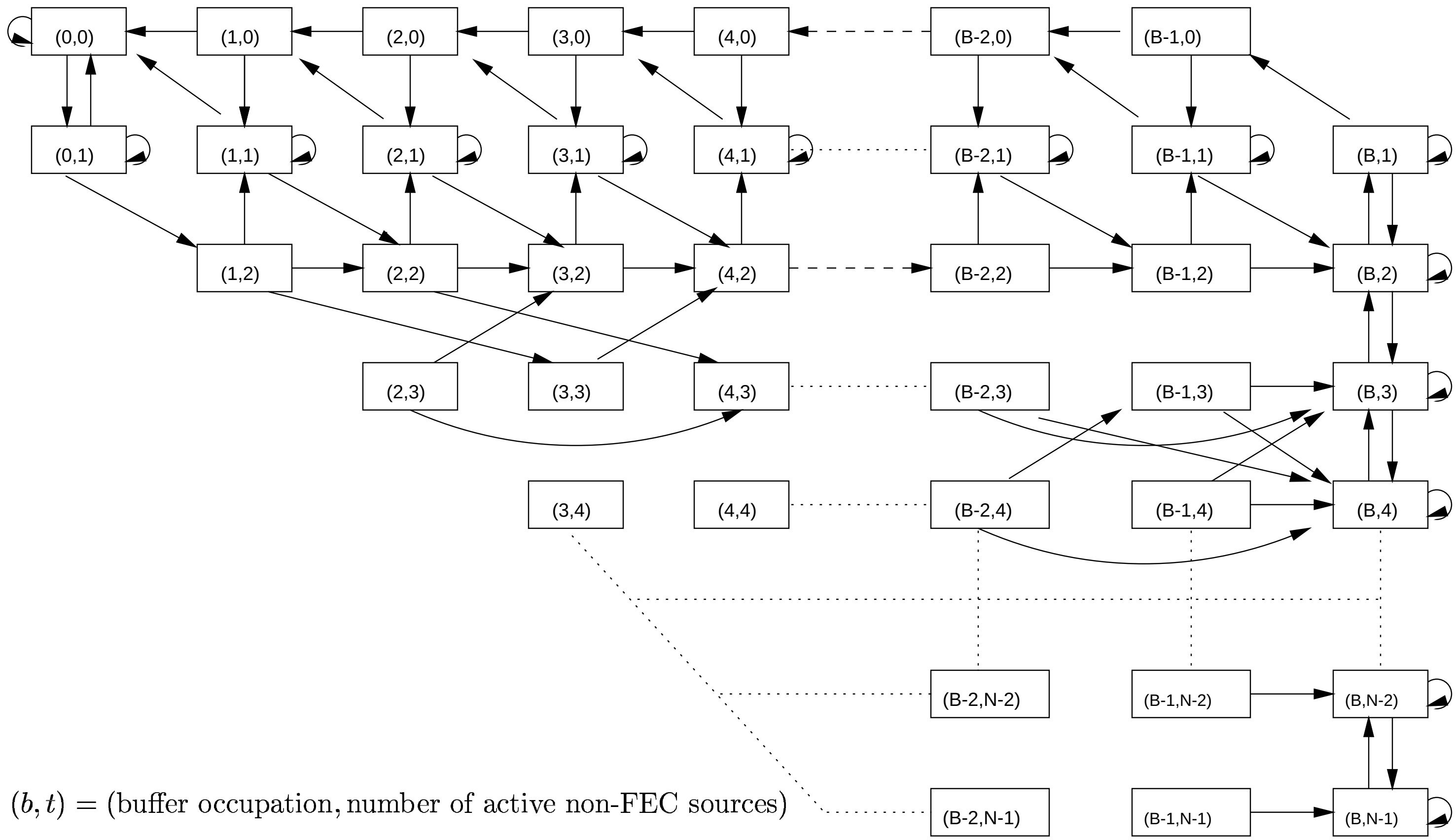
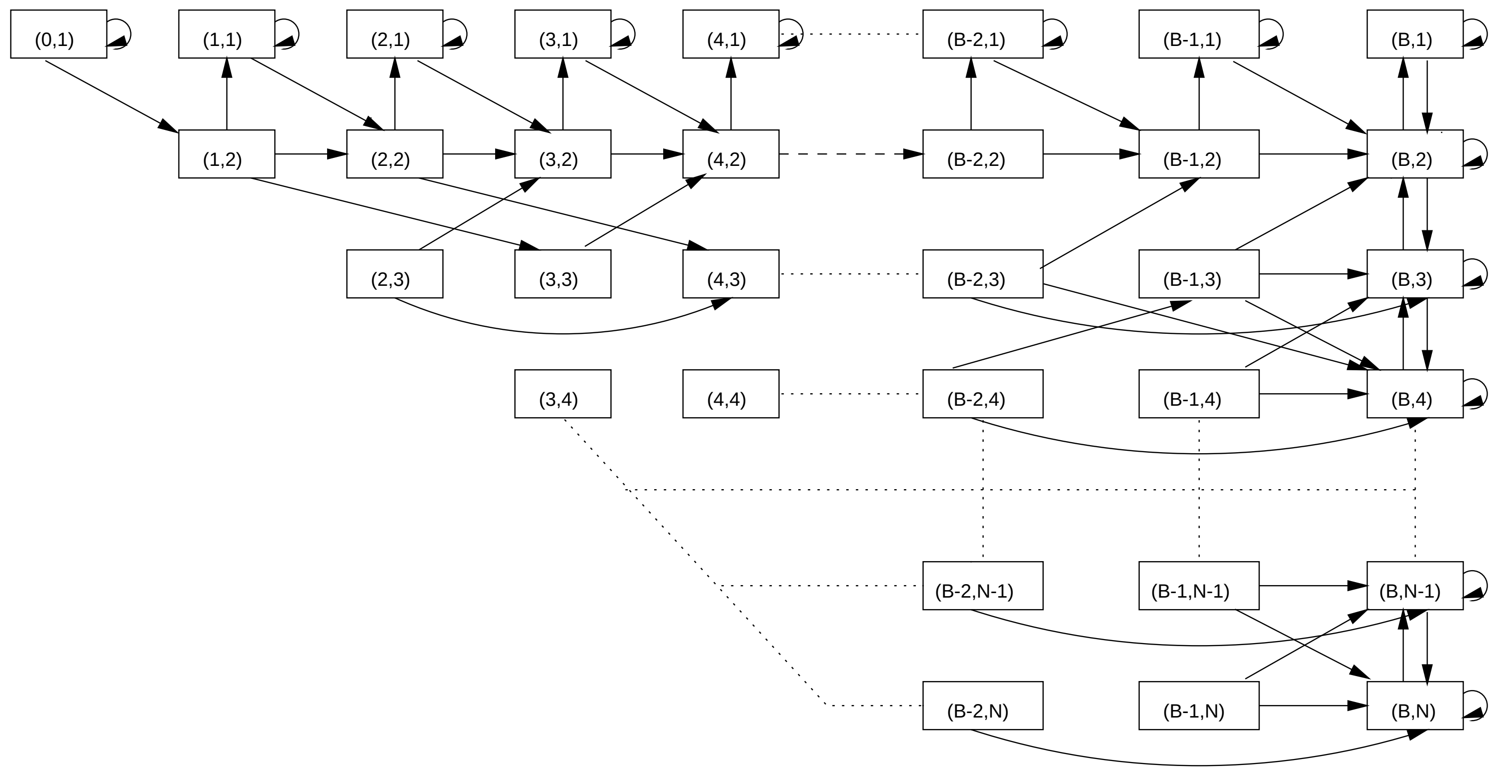


Figure 3: States of $\mathcal{S}_D$ $(t = 0)$.



$$(b, t) = (\text{buffer occupation}, \text{number of active non-FEC sources} + 1)$$

Figure 4: States of $S_S \cup S_L$ $(t = 1)$.

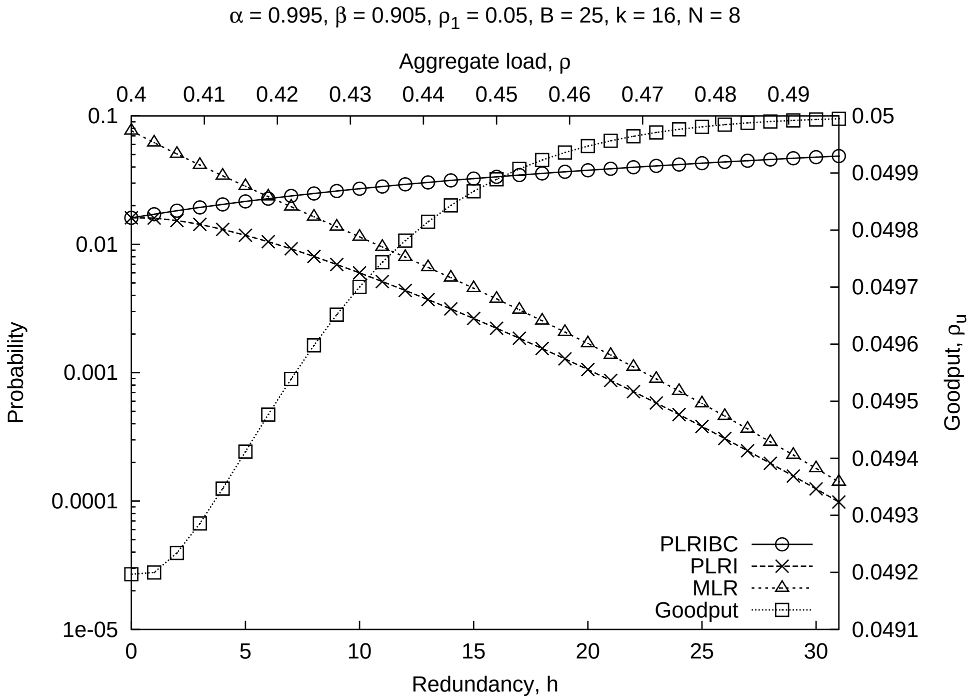


Figure 5: 1 FEC source, medium load.

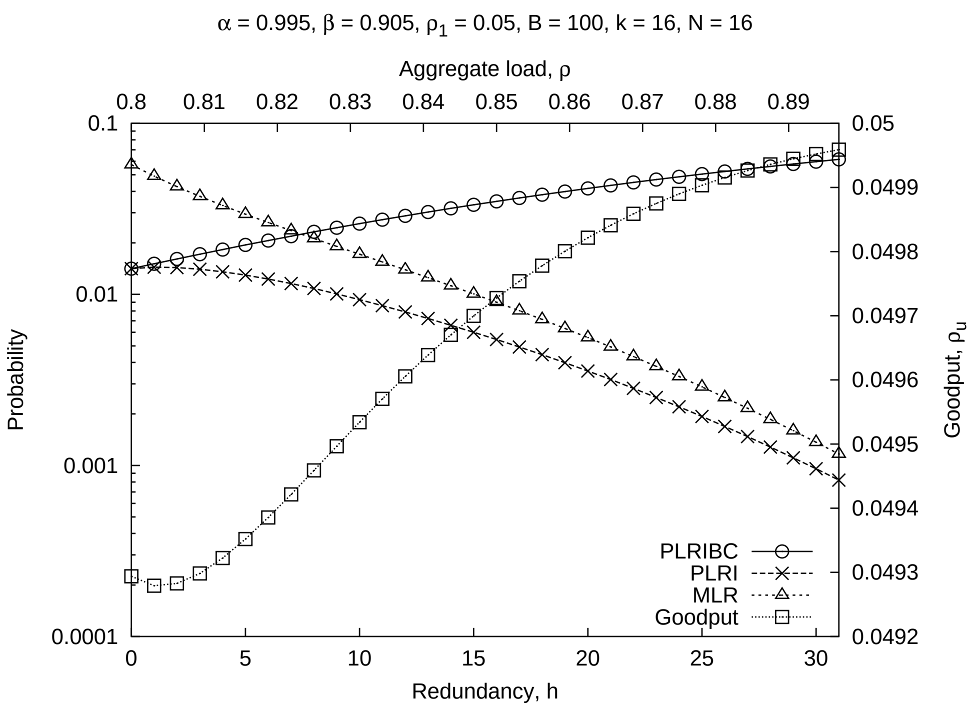


Figure 6: 1 FEC source, high load.

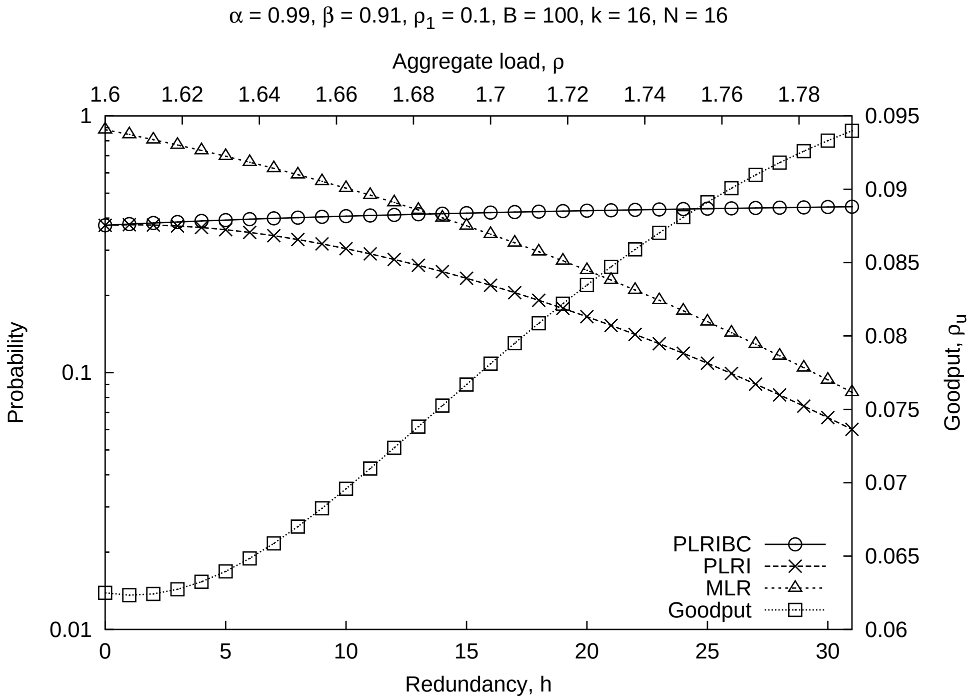


Figure 7: 1 FEC source, overload.

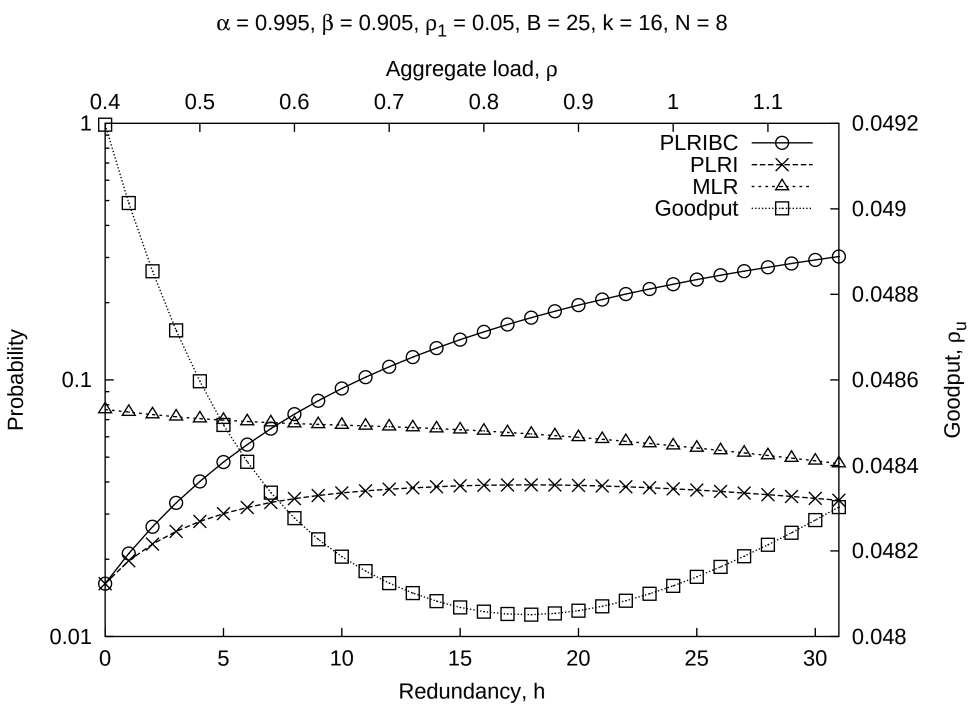


Figure 8: N FEC sources, medium load.

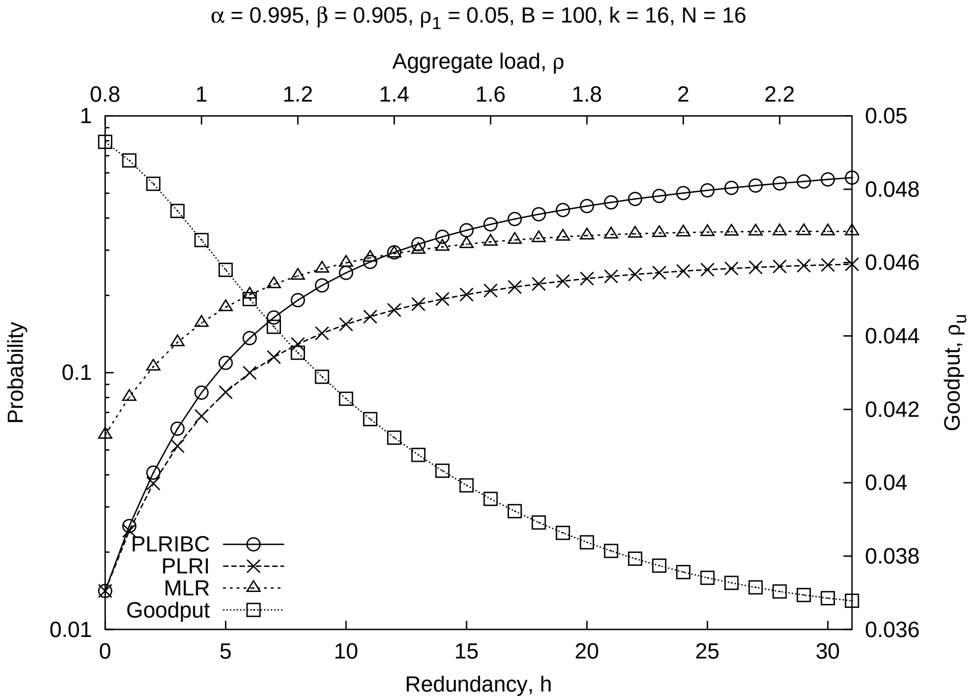


Figure 9: N FEC sources, high load.

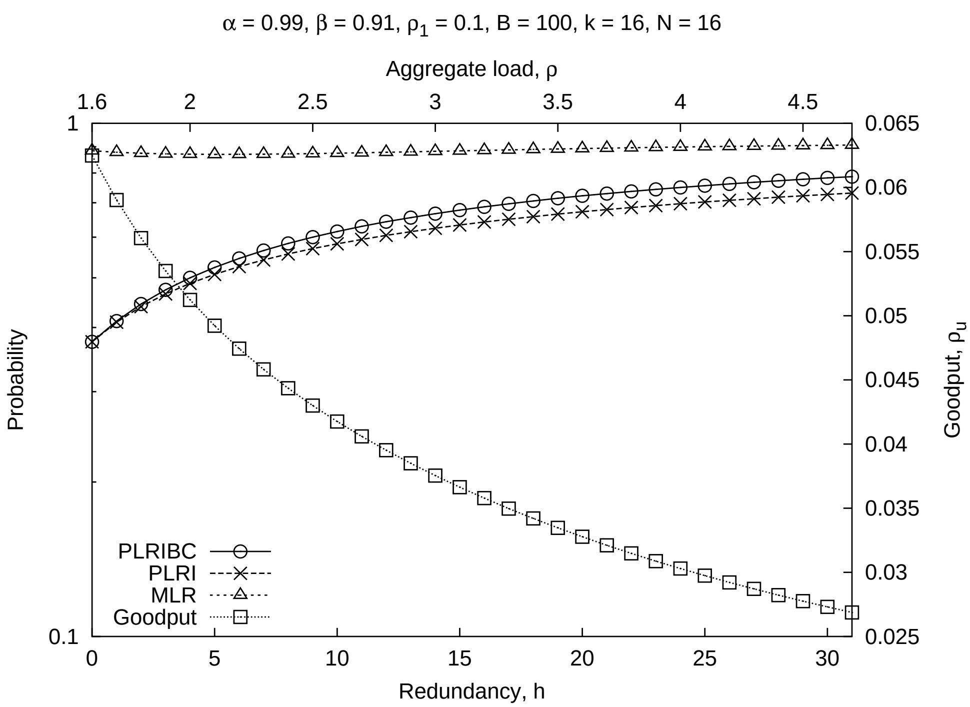


Figure 10: N FEC sources, overload.

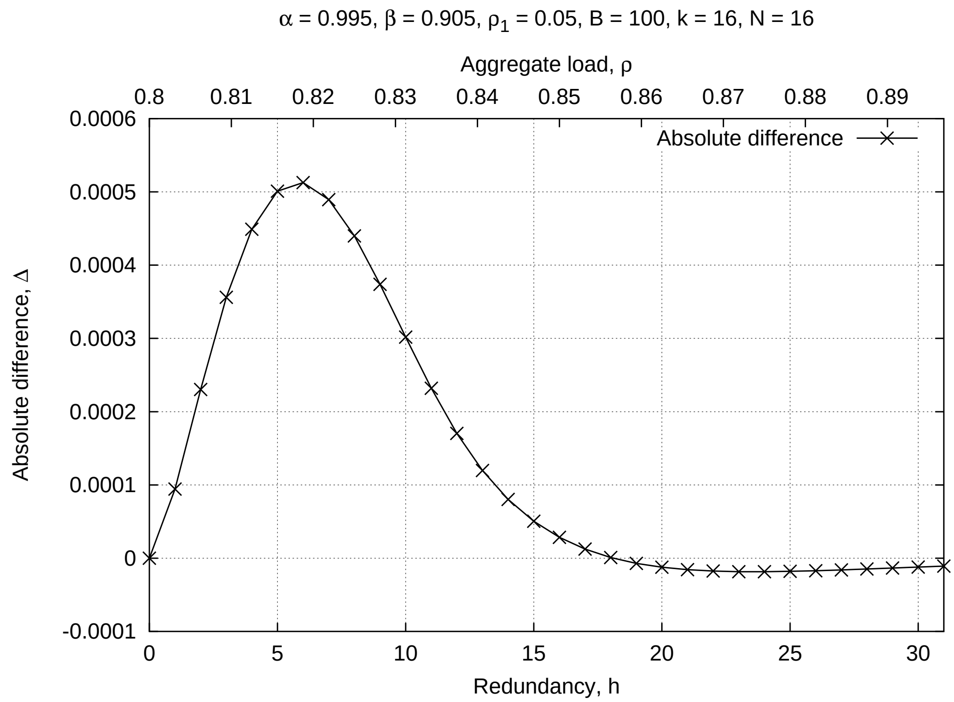


Figure 11: Absolute difference (one FEC source).

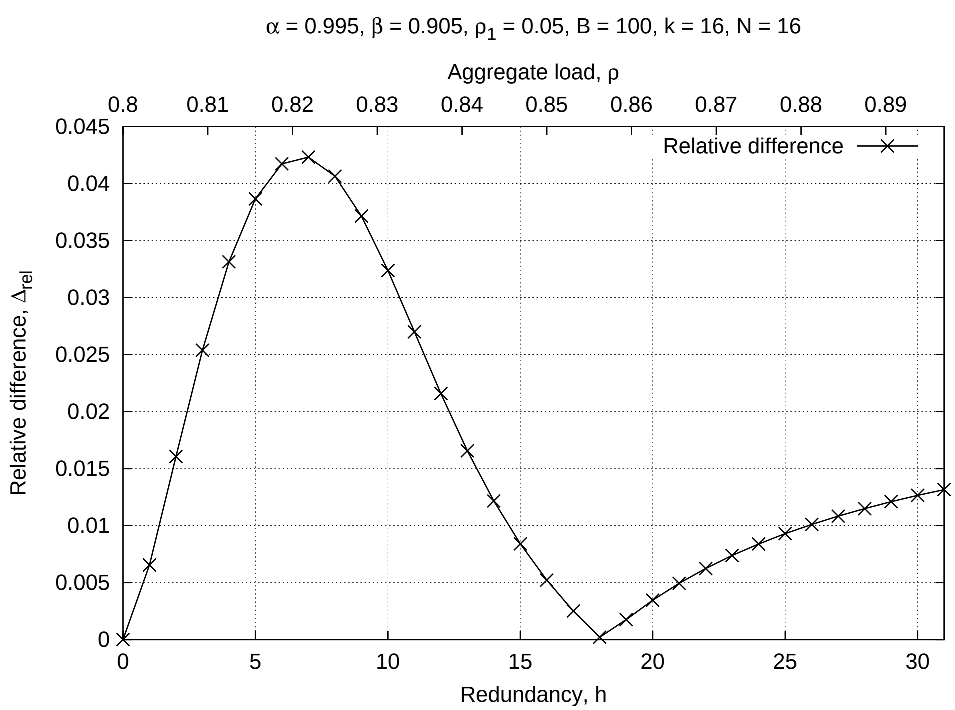


Figure 12: Relative difference (one FEC source).