

Estudo sobre a Concatenação Bidimensional de Códigos LT

Marcelo Corrêa Ramos* e Weiler Finamore*

Resumo—Os códigos LT Bidimensionais, introduzidos em [1], possuem seus símbolos de entrada agrupados em uma matriz, denominada neste trabalho por Matriz LT. Às suas linhas, aplica-se uma codificação LT – no caso, sistemática – e, às colunas da matriz resultante, uma nova codificação LT – convencional. Visando uma melhora de desempenho, a motivação deste trabalho é analisar, neste esquema de codificação em duas dimensões, a influência dos seguintes fatores: (i) tamanho das linhas e colunas da Matriz LT; (ii) valores iniciais dos *overheads* horizontal e vertical e (iii) códigos LT utilizados. É visto que, em matrizes com igual capacidade de símbolos, apresentam melhores resultados aquelas cujo número de linhas é maior. Recomenda-se também, ao final do trabalho, o uso de baixos valores de *overhead* vertical. Conjectura-se a obtenção de melhores resultados com o uso de dois códigos sistemáticos no esquema de codificação LT bidimensional. Todas as simulações foram obtidas para um canal BEC (*Binary Erasure Channel*), que é o canal para o qual os códigos LT foram originalmente criados.

Palavras-Chave—Códigos LT Bidimensionais, Fontanas Digitais, *Binary Erasure Channel* (BEC).

Abstract—Introduced in [1], the Bidimensional LT codes have the input symbols grouped as a matrix (LT Matrix). This symbols are encoded line by line – by a systematic LT code – with the resultant matrix, subsequently, encoded column by column – by a non-systematic LT code. Aiming to improve the performance, our motivation is to analyse the influence, at this bidimensional scheme, of: (i) the size of rows and columns of the LT Matrix; (ii) the initial values of the horizontal and vertical overheads and (iii) the LT codes used. It is shown that, in matrices with the same capacity of symbols, those with more rows achieved the best results. The use of low values of vertical overhead is recommended. On the other hand, the results lead us to conjecture that the use of systematic LT codes at the Bidimensional LT Codes would improve performance. We believe that different combinations of LT codes could improve the results. Simulations were obtained for a BEC channel.

Keywords—Bidimensional LT Codes, Digital Fountains, Binary Erasure Channel (BEC).

I. INTRODUÇÃO

Os códigos LT (*Luby Transform*), propostos por Michael Luby em 2002 [2], foram a primeira implementação do conceito de fontanas digitais (*digital fountains*) com tempos de codificação e decodificação viáveis em sistemas que requeiram a formação de um grande número de símbolos codificados n , a partir de k símbolos de entrada. Por serem baseados em operações *XOR*, estes códigos apresentam altas velocidades de codificação e decodificação, além de oferecerem uma promissora capacidade na correção de erros [3]. Os códigos LT foram originalmente concebidos para canais com apagamento,

BEC (*Binary Erasure Channel*), livres de erros. Em tais canais, consideraremos que os blocos transportados podem ser apagados com uma probabilidade P_a .

Nos códigos LT, cada símbolo codificado pode ser gerado de maneira independente e o conjunto de k símbolos originais de entrada pode ser recuperado, com probabilidade $(1 - \delta)$, a partir de quaisquer $k + O\left(\sqrt{k} \cdot \ln^2\left(\frac{k}{\delta}\right)\right)$ símbolos codificados, com uma média de $O\left(k \cdot \ln\left(\frac{k}{\delta}\right)\right)$ operações-símbolo [2]. O número de símbolos codificados que podem ser gerados a partir dos dados de entrada é potencialmente ilimitado. Por isso, os códigos LT são ditos *rateless* (de taxa versátil). Assim, independentemente do modelo de perdas do canal em uso, o codificador simplesmente gera e envia dados até o decodificador e este envio só cessará quando o número de dados recebido for suficiente para a recuperação da mensagem original. A característica do canal somente influencia no tempo necessário para que os símbolos de saída sejam recebidos pelo decodificador em quantidade suficiente para o sucesso da decodificação. Em qualquer canal com apagamento, os códigos LT são quase ótimos, no sentido de que o decodificador pode recuperar a mensagem original formada por k símbolos de entrada a partir de quaisquer $(1 + \epsilon)k$ símbolos codificados, onde $0 < \epsilon < 1$. Considera-se um código ótimo aquele que consegue recuperar a mensagem original, formada por k símbolos de entrada, a partir de qualquer subconjunto de k símbolos de saída dentre os n gerados [4]. Por serem quase ótimos e por sua eficiência aumentar com o crescimento do comprimento dos dados, os códigos LT são ditos universais [2].

A. Estrutura do Trabalho

Este trabalho descreve, na Seção II, o processo de codificação e de decodificação dos códigos LT, assim como as distribuições de graus utilizadas: Sóliton Robusta Melhorada e Sóliton Robusta Melhorada Truncada (esta última usada com códigos LT sistemáticos). Na Seção III, introduz-se o conceito de códigos LT em duas dimensões e explica-se a codificação 2D e a decodificação 2D. A Seção IV reúne os resultados obtidos e também suas análises. Finalmente, as conclusões do trabalho encontram-se na Seção V.

II. CÓDIGOS LT

No processo de **codificação** dos códigos LT, a mensagem original a ser transmitida consiste de k símbolos de informação, denotados como $s_{i=1}, s_{i=2}, \dots, s_{i=k}$. Em geral, $s_i \in GF(2)^L$, mas por simplicidade considera-se $L = 1$. Cada símbolo codificado t_n é dado por:

* Centro de Estudos em Telecomunicações da PUC-Rio - CETUC, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil. E-mails: marcelocorrearamos@gmail.com, weiler@cetuc.puc-rio.br.

$$t_n = \sum_{\ell=1}^{d_n} s_{i_\ell} \quad (1)$$

onde $d_n \in \{1, k\}$ é uma variável aleatória inteira com distribuição de probabilidade (DP) $\mu(d) = P(d_n = d)$ e, para cada n , $i_\ell \in \{1, k\}$. O símbolo \sum é usado para representar a soma módulo 2 (XOR) dos símbolos s_i . Vale observar que $t_1, t_2, \dots, t_k, \dots \in \{0, 1\}^*$, onde $\{0, 1\}^*$ é o conjunto de todos os blocos binários.

Duas distribuições de probabilidades são usadas nesse processo: uma ($\mu(d)$) para determinar o grau d_n de cada símbolo codificado t_n e outra, uniforme, para escolher os símbolos de entrada *vizinhos* (que entram na formação) do símbolo codificado. O processo de codificação pode ser associado a um grafo biparticionado onde os símbolos de entrada (*nós de mensagem*) estão ligados aos símbolos de saída (*nós de verificação*). A Tabela I ilustra um exemplo do processo de codificação, onde o símbolo \oplus denota a operação XOR.

TABELA I
PROCESSO DE CODIFICAÇÃO

n	Grau (d_n)	Nós vizinhos	t_n
1	2	$s_1 s_2$	$s_1 \oplus s_2 = 1$
2	3	$s_1 s_2 s_4$	$s_1 \oplus s_2 \oplus s_4 = 0$
3	4	$s_1 s_2 s_3 s_5$	$s_1 \oplus s_2 \oplus s_3 \oplus s_5 = 0$
4	4	$s_1 s_3 s_4 s_5$	$s_1 \oplus s_3 \oplus s_4 \oplus s_5 = 1$
5	1	s_5	$s_5 = 1$
6	2	$s_2 s_5$	$s_2 \oplus s_5 = 1$

A título de ilustração, a Fig. 1 mostra o grafo correspondente ao processo de codificação exemplificado na Tabela I. Como terminologia, temos: os símbolos de saída, ou codificados, t_n formam os nós de verificação e os símbolos de entrada, ou de informação, s_i formam os nós de mensagem do grafo. Cada símbolo codificado está conectado, através de arestas, aos símbolos de informação pertencentes ao seu conjunto de nós vizinhos.

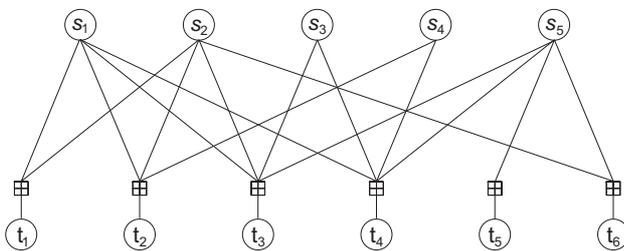


Fig. 1. Grafo resultante da codificação da Tabela I.

Na **decodificação** LT, pressupõe-se que os graus e as conexões são conhecidos. Todos os símbolos de informação são, inicialmente, incógnitas. Um símbolo de informação é dito revelado se, ao longo da decodificação, recebe-se um símbolo codificado de grau unitário que o tem como vizinho. No primeiro passo do processo de decodificação LT, os símbolos codificados com apenas um vizinho são utilizados para revelar seu vizinho único. O conjunto de símbolos de informação

recém revelados é chamado *ripple*. Em cada passo subsequente do processo de decodificação LT, um símbolo de informação no *ripple* é processado, ou seja, ele é removido como vizinho de todos os outros símbolos codificados que o tenham como tal e, posteriormente, todos os símbolos codificados que possuam apenas um vizinho são usados para revelar tal vizinho restante.

O processo termina quando o *ripple* encontrar-se vazio em algum passo. O processo falha quando existe ao menos um símbolo de informação no estado de incógnita ao final do processo e tem sucesso se, no final, todos os símbolos de informação estiverem revelados.

Conforme já mencionado, para cada símbolo codificado t_n é escolhido um grau (número de vizinhos), seguindo-se uma distribuição. Algumas distribuições vêm sendo aplicadas com este intuito. Um projeto de distribuição de graus apropriado assegura que o processo de decodificação LT utilize os símbolos codificados de forma incremental para revelar todos os símbolos de informação. Os objetivos de uma apropriada distribuição de graus são:

- Utilizar lentamente os símbolos codificados enquanto o processo evolui para manter o *ripple* pequeno, de modo a evitar revelação redundante dos símbolos de informação no *ripple* por vários símbolos codificados.
- Não permitir que o *ripple* desapareça antes que todos os símbolos de informação estejam devidamente revelados.

Neste trabalho, são utilizados dois tipos de distribuições de graus: a distribuição Sóliton Robusta Melhorada [5] e a distribuição Sóliton Robusta Melhorada Truncada para códigos LT sistemáticos [7], descritas a seguir.

A. Distribuição Sóliton Robusta Melhorada

Tee, Nguyen *et al.* [5] observaram a existência de algumas distribuições de graus, com probabilidades $P(d_n)$ tão baixas, que o número de símbolos dado pelo produto de $P(d_n)$ e k pode ser menor que 1, indicando a ausência de símbolos com estes graus. Em alguns casos, a distribuição de Luby, denominada de Sóliton Robusta [2], pode levar a um prematuro fracasso na decodificação e uma conseqüente perda de símbolos durante o processo de decodificação, a menos que o número de símbolos redundantes seja elevado. Quando o processo de decodificação é interrompido devido à ausência de símbolos de grau 1, precisa-se receber mais símbolos para a recuperação de todos os símbolos originais.

A proposta de Tee, Nguyen *et al.*, portanto, é melhorar o comportamento da distribuição Sóliton Robusta no caso em que $P(d_n) \cdot k < 1$, através da introdução de um fator extra (ν) cuja finalidade é criar uma distribuição de grau mais benéfica. Em referência à Sóliton Robusta de Luby, esta distribuição é denominada Sóliton Robusta Melhorada (DSRM) e é definida por:

$$\mu(d) = \frac{(\rho(d) + \tau(d))}{\beta} + \nu, \quad (2)$$

onde

$$\rho(d) = \begin{cases} \frac{1}{k} & \text{para } d = 1, \\ \frac{1}{d \cdot (d-1)} & \text{para } d = 2, 3, \dots, k \end{cases} \quad (3)$$

e

$$\tau(d) = \begin{cases} \frac{R}{d \cdot k} & \text{para } 1 \leq d \leq \frac{k}{R} - 1, \\ \frac{R \cdot \ln(\frac{R}{\delta})}{k} & \text{para } d = \frac{k}{R}, \\ 0 & \text{para } d = \frac{k}{R} + 1, \dots, k, \end{cases} \quad (4)$$

com $R = c \cdot \ln(k/\delta) \cdot \sqrt{k}$, sendo a constante $c > 0$. O fator ν é definido por:

$$\nu = \sum \mu(d_n) \cdot k, \quad (5)$$

onde d_n representa o termo grau- n da distribuição, satisfazendo as seguintes condições:

$$\begin{cases} \left(\frac{1}{d \cdot (d-1)} + \frac{R}{k \cdot d} \right) \cdot \frac{k}{\beta} < 1 & \text{para } 2 \leq d \leq \left(\frac{k}{R} - 1 \right), \\ \left(\frac{1}{d \cdot (d-1)} \right) \cdot \frac{k}{\beta} < 1 & \text{para } \left(\frac{k}{R} + 1 \right) \leq d \leq k. \end{cases} \quad (6)$$

Têm-se também, da Equação 2, o fator de normalização $\beta = \sum_d [\rho(d) + \tau(d)]$, garantindo que as somas das probabilidades seja unitária.

Na Equação 2, o parâmetro ν é determinado obedecendo-se ao conjunto de desigualdades dado pela Equação 6, no intuito de maximizar a frequência relativa símbolos de grau 1. Ou seja, todos aqueles graus d_n que satisfazem a Equação 6 têm redefinidas as suas probabilidades de distribuição a zero e a soma de todas essas probabilidades de distribuição são adicionadas à distribuição de probabilidade de grau 1.

B. Distribuição Sóliton Robusta Melhorada Truncada para Códigos LT Sistemáticos

Na distribuição Sóliton Robusta, há duas condições a serem satisfeitas por todos os símbolos de entrada a fim de serem recuperados no decodificador. Primeiramente, o número de símbolos codificados (n) deve satisfazer $n \geq k + 2 \cdot \ln(\frac{R}{\delta}) \cdot R$. Também, o número de símbolos que possuem o mais alto grau deve obedecer a $d \geq \frac{k}{R}$ [2] e [8]. Nguyen, Yang *et al.* ao desenvolverem um código LT sistemático em [7], viram que estas condições continuavam válidas para a decodibilidade de tais códigos com taxas maiores que $T = \frac{1}{2+\epsilon}$, onde $\epsilon = 2 \cdot \ln(\frac{R}{\delta})$ é o *overhead* do código LT original. Entretanto, para taxas menores que $T = \frac{1}{2+\epsilon}$, essas condições não são suficientes, já que as densidades de ambas as matrizes, geradora e de paridade, dos códigos LT sistemáticos são baixas.

No intuito de aumentar essas densidades, a distribuição de graus usada para gerar a matriz de paridade dos códigos LT sistemáticos, denominada distribuição Sóliton Robusta Melhorada Truncada (DSRMT), foi redefinida da seguinte forma [6]:

$$\Omega(d) = \begin{cases} \frac{1}{\beta'} \left[1 + \frac{R}{k} + \nu \right] & \text{para } d = \gamma, \\ \frac{\gamma}{\beta'} \left[\frac{1}{d \cdot (\frac{d}{\gamma} - 1)} + \frac{R}{k} \frac{1}{d} \right] & \text{para } d = 2\gamma, 3\gamma, \\ & \dots, \frac{\gamma \cdot k}{R} - 1, \\ \frac{R}{\beta' \cdot k} \left[\log\left(\frac{R}{\delta}\right) + \frac{1}{\left(\frac{k}{R} - 1\right)} \right] & \text{para } d = \frac{\gamma \cdot k}{R}, \\ 0 & \text{para } d > \frac{\gamma \cdot k}{R} \text{ e } d = 1, \end{cases} \quad (7)$$

onde k é o número total de símbolos de informação originais e R é o número de símbolos com um grau específico γ , que satisfaz a condição de [8] $R \equiv c \cdot \ln\left(\frac{k}{\delta}\right) \cdot \sqrt{k}$. Além do mais, ν representa o fator extra que garante a decodibilidade da distribuição Sóliton Robusta Melhorada, já visto na Equação 5. Ainda em referência à Equação 7, temos $\beta' = \sum_d [\rho(d) + \tau(d) + \nu(\gamma)]$, onde γ é um número inteiro maior que 1. Mantendo-se o grau máximo em $d_{max} = \frac{\gamma \cdot k}{R}$, garantimos que os símbolos originais de entrada serão representados no grupo de símbolos sistemáticos codificados, pelo menos γ vezes [2], [8] e [9].

A média dos graus dos símbolos sistemáticos codificados é dada pela seguinte equação:

$$D = \sum_d \frac{d \cdot (\rho(d) + \tau(d) + \nu(d))}{\beta'} + 1. \quad (8)$$

III. CÓDIGOS LT BIDIMENSIONAIS

Os códigos LT Bidimensionais, propostos por Sanchez em [1], têm seus símbolos de entrada rearranjados em forma de uma matriz, como mostra a Fig. 2. Tais códigos consistem da concatenação de duas codificações, uma na horizontal e outra na vertical.

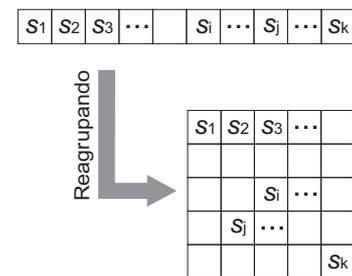


Fig. 2. Reagrupamento bidimensional dos símbolos de entrada s_i .

Quando os símbolos de entrada estão rearranjados em uma matriz, aplica-se um código LT linha a linha. Por trabalhar horizontalmente, denominamos este código de LT horizontal (LT_H). O código LT_H introduz um *overhead* (ϵ_H) em cada linha e então, aplicamos um outro código LT (LT_V), agora verticalmente, coluna a coluna. O código LT_V , por sua vez, introduz um *overhead* (ϵ_V) em cada coluna. O *overhead* total do sistema é dado por $(1 + \epsilon_{TOTAL}) = (1 + \epsilon_H)(1 + \epsilon_V)$. A Fig. 3 ilustra este processo.

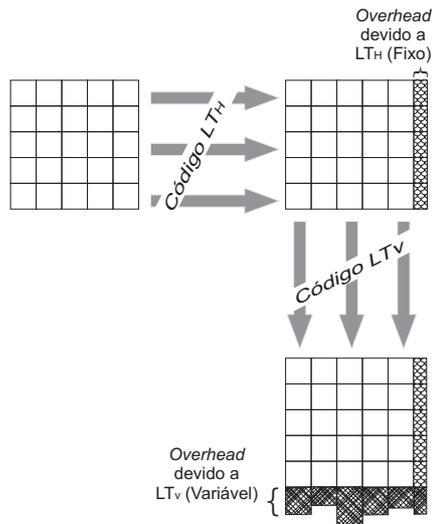


Fig. 3. Processo de codificação bidimensional.

Neste trabalho, por se propor a analisar os fatores influentes na codificação LT bidimensional proposta em [1], é utilizado como LT_H um código LT sistemático com distribuição Sóliton Robusta Melhorada Truncada (DSRMT) e, como LT_V , um código LT com distribuição Sóliton Robusta Melhorada (DSRM).

O processo de **codificação 2D** dos códigos LT Bidimensionais consiste de duas codificações unidimensionais tradicionais concatenadas, uma no sentido horizontal e outra no sentido vertical da Matriz LT.

No processo de **decodificação 2D**, primeiramente é realizada a decodificação vertical e, depois, a horizontal. Em caso de falha, o decodificador requisita que mais símbolos sejam enviados (aumentando o *overhead*) e o processo é repetido. Isto ocorre até que a decodificação obtenha sucesso. Na implementação deste trabalho, os símbolos adicionais requeridos pelo decodificador são enviados pelo codificador LT_V . Assim, conforme a Fig. 3, o *overhead* variável é o vertical.

IV. RESULTADOS

Em [1], Sanchez usou em suas simulações uma Matriz LT de tamanho [1000x1000] e valores de *overheads* horizontal e vertical, respectivamente, de $\epsilon_H = 5\%$ e $\epsilon_V = 24\%$, totalizando um *overhead* total de $\epsilon_{TOTAL} \approx 30\%$.

Neste trabalho, objetivamos fazer um estudo da influência do tamanho desta Matriz LT, dos valores de ϵ_H e ϵ_V e também dos códigos LT utilizados, no desempenho dos códigos LT Bidimensionais. Nesta seção são apresentados os resultados deste estudo. As simulações foram realizadas em um programa desenvolvido em linguagem MATLAB. Todas as simulações foram obtidas para canal BEC e, por simplificação, um símbolo foi composto por apenas um *bit*.

Buscando investigar a influência do tamanho da Matriz LT, comparou-se a Matriz LT [1000x1000] utilizada em [1] com a matriz [2000x500]. Como pode-se observar, ambas têm igual capacidade de símbolos (1Mega). Com o valor do *overhead*

horizontal fixo em 3%, 5% ou 10%, variou-se o *overhead* vertical inicial e obteve-se o *overhead* total necessário para uma decodificação com sucesso. As simulações foram feitas em canal com probabilidade de apagamento $P_a = 5\%$.

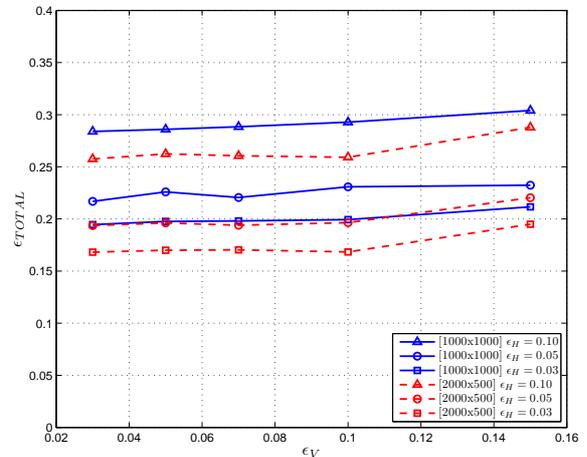


Fig. 4. ϵ_{TOTAL} requerido, em função de ϵ_H e ϵ_V , para as seguintes Matrizes LT: [1000x1000], [2000x500]. Parâmetros utilizados: $P_a = 5\%$, $c = 0.03$, $\rho = 0.1$ e $\gamma = 6; 5$, respectivamente com as matrizes.

A Fig. 4 mostra estes resultados. A matriz [2000x500] requereu um *overhead* de 2% a 4% menor que a matriz [1000x1000], nas diversas combinações de valores de ϵ_H e ϵ_V . Portanto, conclui-se que, para matrizes LT com igual capacidade de símbolos, quanto maior o número de linhas, menor será o *overhead* requerido para uma decodificação com sucesso.

Na Fig. 4 comparou-se as matrizes [1000x1000] e [2000x500], de capacidades iguais a 1Mega símbolos. Nas Figs. 5 e 6 essas matrizes são comparadas, respectivamente, às matrizes [1000x500] e [2000x250], ambas de capacidade igual a 0.5Mega símbolos. Nas duas comparações, independentemente do valor de ϵ_H utilizado, a diminuição da capacidade das matrizes LT de 1Mega para 0.5Mega, através da diminuição do número de colunas, não acarreta nenhum efeito considerável no ϵ_{TOTAL} requerido para uma decodificação com sucesso.

Tendo em vista que os resultados das matrizes [1000x1000] e [2000x500] foram, respectivamente, praticamente idênticos aos das matrizes [1000x500] e [2000x250], resolveu-se, por simplicidade, fazer uso das matrizes com capacidade de 0.5Mega símbolos.

Como em nossas simulações, o código LT_H é diferente do LT_V , testou-se a influência do peso de cada um dos códigos no processo de decodificação bidimensional. É sabido que os códigos LT melhoram seu desempenho com o aumento do tamanho do bloco de informação. Portanto, o código LT_H tem maior peso quando o número de colunas é maior que o número de linhas da matriz. Quando o inverso ocorre, o código LT_V passa a ter um maior peso no processo. A Fig. 7 mostra as matrizes [2000x250], [1000x500], [500x1000] e [250x2000]. Essas matrizes permitem a análise da diminuição do número

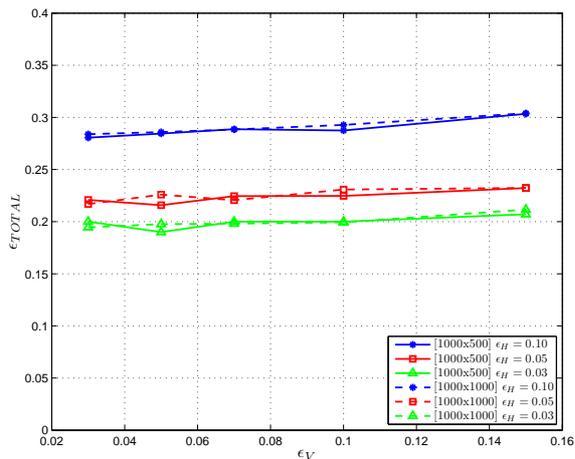


Fig. 5. ϵ_{TOTAL} requerido, em função de ϵ_H e ϵ_V , para as seguintes Matrizes LT: [1000x500] e [1000x1000]. Parâmetros utilizados: $P_a = 5\%$, $c = 0.03$, $\rho = 0.1$ e $\gamma = 5; 6$, respectivamente com as matrizes.

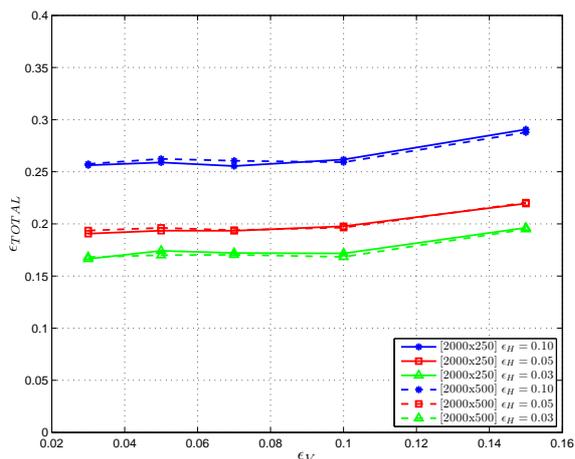


Fig. 6. ϵ_{TOTAL} requerido, em função de ϵ_H e ϵ_V , para as seguintes Matrizes LT: [2000x250] e [2000x500]. Parâmetros utilizados: $P_a = 5\%$, $c = 0.03$, $\rho = 0.1$ e $\gamma = 3; 5$, respectivamente com as matrizes.

de linhas (2000 \rightarrow 1000 \rightarrow 500 \rightarrow 250) em contrapartida ao aumento do número de colunas (250 \rightarrow 500 \rightarrow 1000 \rightarrow 2000) e vice-versa.

A matriz [2000x250] foi a que requereu menor *overhead* para uma decodificação com sucesso. Os piores resultados foram da matriz [250x2000]. Estes resultados, alinhados aos das Figs. 4 a 6, mostram que os casos onde o LT_V tem maior peso são melhores. Para uma determinada combinação dos valores dos *overheads*, à medida em que o peso do LT_H aumenta – aumento do número de colunas e diminuição do número de linhas – têm-se um aumento de mais de 10% no valor do ϵ_{TOTAL} requerido.

Na Fig. 8, outro fator comprova a pouca eficiência do LT_H utilizado na codificação bidimensional. Nela, algumas combinações de ϵ_H e ϵ_V (iniciais) são comparadas para uma

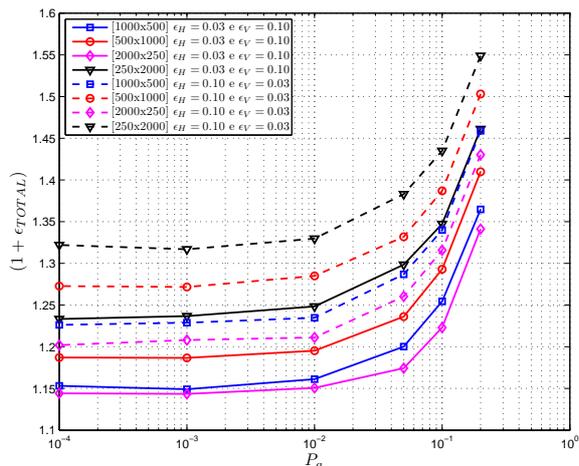


Fig. 7. ϵ_{TOTAL} requerido, em função de ϵ_H e ϵ_V , para as seguintes Matrizes LT: [1000x500], [500x1000], [2000x250] e [250x2000]. Parâmetros utilizados: $c = 0.03$, $\rho = 0.1$ e $\gamma = 5; 6; 3; 13$, respectivamente com as matrizes.

Matriz LT [1000x500]. Têm-se que, fixando-se $\epsilon_V = 3\%$, o aumento de $\epsilon_H = 10\%$ para $\epsilon_H = 20\%$ acarreta um incremento de aproximadamente 15% no ϵ_{TOTAL} requerido para uma decodificação com sucesso, em qualquer faixa de P_a . Num caso extremo, onde $\epsilon_H = 0\%$, ou seja, sem codificação horizontal – uma vez que o LT_H utilizado é sistemático –, o ϵ_{TOTAL} requerido é cerca de 14% mais baixo que o caso onde $\epsilon_H = 10\%$. Por outro lado, ao fixar $\epsilon_H = 3\%$, observa-se que o aumento de $\epsilon_V = 0\%$ para $\epsilon_V = 20\%$ leva a um acréscimo de 10% no valor de ϵ_{TOTAL} nas regiões de boa qualidade do canal ($P_a < 1\%$). Para canais com altas probabilidades de apagamento ($P_a > 10\%$), este acréscimo é apenas de 3%. Recomenda-se, portanto, o uso de valores de ϵ_V iniciais baixos (até 10%).

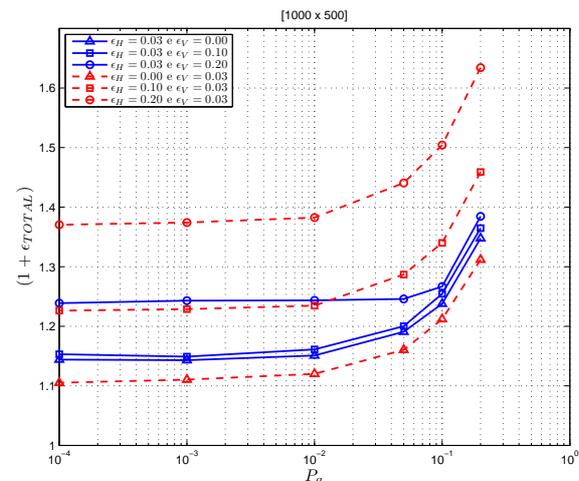


Fig. 8. ϵ_{TOTAL} requerido, em função de P_a , para a Matriz LT [1000x500]. Parâmetros utilizados: $c = 0.03$, $\rho = 0.1$ e $\gamma = 5$.

A Fig. 9 mostra a Taxa de Apagamento de Símbolo (TAS)

em função da qualidade do canal ($1 - P_a$) para diversas Matrizes LT, de mesma capacidade de símbolos (0.5Mega). Por Taxa de Apagamento de Símbolo, entenda-se o percentual de símbolos que não foram revelados ao final do processo de decodificação. No caso destas simulações, os *overheads* horizontal e vertical foram fixados em $\epsilon_H = 0.05$ e $\epsilon_V = 0.14$ ou 0.24, originando um *overhead* inicial – e, no caso, também final – de $\epsilon = 0.20$ ou 0.30, respectivamente. No momento em que o processo falha, a decodificação é finalizada e a TAS contabilizada. Neste caso, não há o requerimento nem o acréscimo de símbolos adicionais.

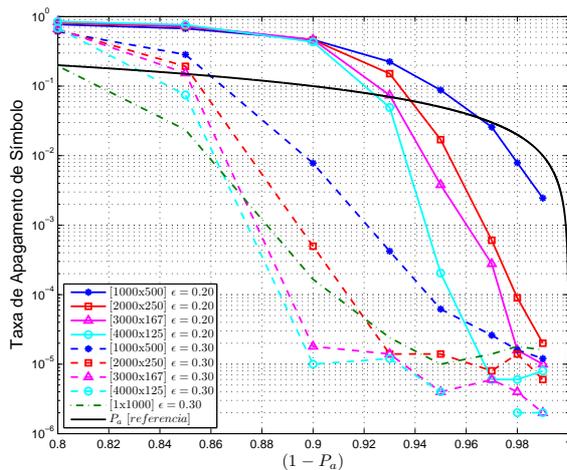


Fig. 9. Taxa de Apagamento de Símbolo em função da probabilidade de apagamento do canal e do ϵ_{TOTAL} , para as seguintes Matrizes LT: [1000x500], [2000x250], [3000x167] e [4000x125]. Parâmetros utilizados: $c = 0.03$, $\rho = 0.1$ e $\gamma = 5; 3; 2; 2$, respectivamente com as matrizes.

Claramente vê-se na Fig. 9 que o aumento do *overhead* melhora consideravelmente o desempenho, em todas as Matrizes LT analisadas. Percebe-se também que o desempenho melhora com o aumento do número de linhas. Também na Fig. 9, têm-se a curva da matriz [1x1000], com $\epsilon = 0.30$, para efeito de comparação com os esquemas bidimensionais. Para manter a mesma capacidade dos demais esquemas utilizados, esta matriz unidimensional foi simulada 1000 vezes e então, uma média de sua TAS foi calculada. Para valores menores que $P_a = 13\%$, alguns esquemas bidimensionais ([2000x250], [3000x167] e [4000x125]) superam a curva unidimensional, para $\epsilon = 0.30$. Esses resultados mostram que a proposta da codificação bidimensional LT é válida e que os seus estudos devem ter continuidade.

V. CONCLUSÕES

Neste trabalho fez-se um estudo da influência das Matrizes LT, dos valores iniciais de *overheads* horizontal e vertical e também dos códigos LT utilizados na codificação LT bidimensional introduzida em [1].

Concluiu-se, para a configuração de códigos LT Bidimensionais utilizada que, para matrizes com igual capacidade de símbolos, é melhor usar aquelas com maior número de

linhas. Matrizes com capacidade de 0.5Mega símbolos requerem a mesma quantidade de símbolos adicionais, para uma decodificação com sucesso, que matrizes com capacidade de 1Mega símbolos. Portanto, neste caso, aconselha-se por simplicidade o uso daquelas de menor capacidade. No caso dos *overheads*, observou-se que, em situações com baixa qualidade de canal ($P_a > 10\%$), o valor inicial do ϵ_V tem pouca influência no desempenho do sistema, recomendando-se um baixo valor do mesmo (até 10%). Já o aumento do ϵ_H inicial, piora significativamente o desempenho da codificação LT bidimensional utilizada, para qualquer valor de P_a .

Foi detectado, ainda, um aumento de mais de 10% no valor do ϵ_{TOTAL} requerido, com o aumento do peso do LT_H em relação ao LT_V . Espera-se que uma outra combinação de códigos LT, especialmente o uso de dois códigos LT sistemáticos, traga benefícios ao sistema LT bidimensional.

Por fim, em sistemas sem requerimento de símbolos adicionais na decodificação, viu-se que a codificação LT bidimensional melhora o desempenho do sistema num canal com apagamento. Quanto maior for o *overhead* utilizado, combinado ao uso de uma Matriz LT adequada, menor é a Taxa de Apagamento de Símbolos (TAS). Em relação a um esquema unidimensional de igual capacidade, alguns esquemas bidimensionais propostos apresentaram melhores resultados, à medida que a qualidade do canal melhora.

Conjetura-se que resultados melhores podem ser obtidos com os códigos LT Bidimensionais e outras implementações estão sendo investigadas.

AGRADECIMENTOS

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) por financiar este trabalho.

REFERÊNCIAS

- [1] SANCHEZ PAIBA, F.. **Códigos Fontanais Bidimensionais para Canais com Apagamento**. Dissertação de Mestrado, Pontifícia Universidade Católica do Rio de Janeiro / PUC-Rio, Jul.2008.
- [2] LUBY, M.. **LT Codes**. Proceedings of the 43rd Annual IEEE Symp. on Foundation of Comp. Sc., p. 271–280, Nov. 2002.
- [3] SASAKI, C.; HASEGAWA, T.; KOBAYASHI, S.. **On Unicast Based Recovery For Multicast Content Distribution Considering XOR-FEC**. Asia-Pacific Conference on Communications, October 2005.
- [4] MAYMOUNKOV, P.; MAZIERES, D.. **Rateless Codes And Big Downloads**. Proceedings of 2nd International Workshop on Peer-to-Peer System, 2003.
- [5] TEE, R.S.Y.; NGUYEN, T.D.; YANG, L.L.; HANZO, L.. **Serially Concatenated Luby Transform Coding And Bit-Interleaved Coded Modulation Using Iterative Decoding For The Wireless Internet**. IEEE 63rd Vehicular Technology Conference, VTC 2006-Spring, Vol. 1, p. 22–26, 2006.
- [6] NGUYEN, T.D.; YANG, L.L.; HANZO, L.. **An Optimal Degree Distribution Design And A Conditional Random Integer Generator For The Systematic Luby Transform Coded Wireless Internet**. WCNC, 2008.
- [7] NGUYEN, T.D.; YANG, L.L.; HANZO, L.. **Systematic Luby Transform Codes And Their Soft Decoding**. 2007 IEEE Workshop on Signal Processing Systems, p. 67–72, Oct. 2007.
- [8] MACKAY, D. J. C.. **Fountain Codes**. Cavendish Laboratory, University of Cambridge, Oct. 2004, em <http://www.inference.phy.cam.ac.uk/mackay>.
- [9] SHOKROLLAHI, A.. **Raptor Codes**. IEEE Transactions on Information Theory, p. 2551–2567, Jun. 2006.