Avanços em Códigos Fontanais

WEILER A. FINAMORE
Pontifícia Universidade Católica do Rio de Janeiro, Centro de Estudos em Telecomunicações (CETUC/PUC-Rio), 22450-900, Rio de Janeiro - RJ, Brasil. E-address: weiler@cetuc.puc-rio.br.

***Resumo*— Os códigos fontanais introduzidos recentemente representam uma importante contribuição para o transporte de informação através de canais com apagamento. O presente trabalho, apresenta uma descrição dos códigos-LT e de um uma modificação (encurtameno) deste que apresenta um desempenho melhorado.**

***Abstract*— The recently introduced fountain codes are an important tool to send information trough erasure channels. In this paper a description of the Luby Codes is presented together with a code modification (shortening) that enders a better performance.**

I. INTRODUÇÃO

Os códigos fontanais introduzidos recentemente representam uma importante contribuição para o transporte de informação através de canais com apagamento — o BEC (*Binary erasure channel*) em particular. Com pouco mais de dez anos desde o seu surgimento esta técnica impulsionou diversas variações (códigos Tornado, LT, Raptor as mais proeminentes) com ampla utilização prática. O presente trabalho, uma "caminhada" pela trilha dos códigos fontanais, será apresentado com vistas, não apenas de revelar suas múltiplas derivações e aplicações, mas com o propósito de também enfatizar que conhecimentos e tecnologias tão novas como esta têm origem em conceitos básicos e que devem ser compreendidos em sua totalidade. Aceitar o hoje consensual fato de que o estudo do problema da transmissão de informação produzida por uma única fonte a ser enviada a um único destinatário não conduzirá a saltos de progresso significativos não significa deixar de lado o estudo dos fundamentos ou de técnicas fundamentais — com esta motivação iremos apresentar uma discussão simples acerca dos canais BEC e das soluções para transmitir informação através de um canal como este. Esta discussão será acompanhada de um descrição dos códigos-LT e de uma variação deste que apresenta um desempenho melhorado. Conclui-se com uma discussão sobre diversas aplicações e variações destes códigos.

II. CANAL BEC

A transmissão de informação de um ponto do espaço a outro é realizada através de um meio de transmissão não-determinístico, modelado matematicamente por um sistema aleatório. Em nossa discussão vamos considerar sistemas de comunicação digital tal que o modelo matemático do meio de transmissão é um sistema aleatório, denominado *canal*, que encontra em sua entrada, cada vez que é usado, um símbolo pertencente a um conjunto discreto (denominado *alfabeto de entrada* do canal) e em sua saída, também um símbolo pertencente a conjunto discreto (denominado *alfabeto de saída* do canal) que se relaciona probabilisticamente ao símbolo de entrada através de uma probabilidade condicional. Tendo em vista que os *códigos-LT* (*Luby-Transform Codes*) foram inicialmente propostos para proteger dados a serem transmitidos através de um *Erasure Channel* vamos restringir a discussão a tais canais — em particular, por simplidade, a discussão se restringirá a um BEC. O BEC é um canal em que o alfabeto de entrada $\mathcal{A} = \{a_1, a_2\}$ é binário, e o alfabeto de saída, ternário, é $\mathcal{B} = \{b_1, b_2, b_3\}$ — neste canal, se em determinado instante de tempo i o símbolo de saída é $y_i = \beta_1$ (ou $y_i = \beta_2$) pode-se afirmar, com probabilidade igual a 1, que o símbolo de entrada é $x_i = a_1$ (ou $x_i = \alpha_2$) já, se o símbolo de saída é $y_i = \beta_3$, tem-se que a probabilidade do símbolo transmitido ter sido $x_i = a_1$ é $P[X_i = a_1] = 0,5$ (e, de forma semelhante, $P[X_i = 1] = 0,5$). Um canal (dito sem-memória e incariante no tempo) é caracterizado probabilisticamente por uma matriz de probabilidades (denominada *matriz probabilidade de transição*) — o BEC, por exemplo, é modelado como um sistema probabilístico onde a entrada que será observada em um dado instante i é uma v.a. (variável aleatória) X pertencente ao alfabeto de entrada do canal $\mathcal{A} = \{a_1, a_2\}$ e, a saída é uma v.a. $Y \in \mathcal{B} = \{b_1, b_2, b_3\}$. A matriz de probabilidade de transição, tem dimensões 2×3 e cada elemento desta matriz é representado por $q_{k|j} = P[Y = b_k \,|\, X = b_j]$, $(a_j, b_k) \in \mathcal{A} \times \mathcal{B}$ (omite-se o índice i ao se referir à v.a. tendo em vista a estacionariedade do processo). Em geral refere-se ao símbolo de saída b_2 como um *apagamento* já que ao se observar este símbolo na saída do canal nada se pode dizer com relação ao símbolo que foi enviado, contrariamente ao recebimento de outro símbolo, (se $y = b_1$, por exemplo, pode-se afirmar, com probabilidade um, que o símbolo correspondentemente enviado é $x = a_1$). Um BEC(ϵ), com $P[Y = b_2 | X = a_j] = \epsilon, j \in \{1, 2\}$, pode ser caracterizado uzando-se um diagrama conforme ilustra a Fig.1. Observa-se nesta figura que todas as informações necessárias

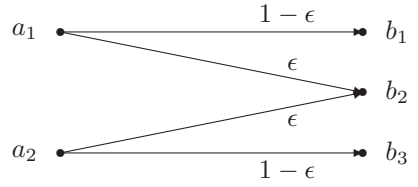

Fig. 1. BEC(ϵ) (canal binário, com apagamento).

para caracterizar probabilisticamente o meio de transmissão estão presentes. A matriz probabilidade de transição

$$\mathbf{Q} = \begin{pmatrix} q_{1|1} & q_{2|1} & q_{3|1} \\ q_{1|2} & q_{2|2} & q_{3|2} \end{pmatrix} = \begin{pmatrix} 1-\epsilon & \epsilon & 0 \\ 0 & \epsilon & 1-\epsilon \end{pmatrix} \quad (1)$$

caracteriza probabilisticamente um BEC(ϵ).

III. Tx através de BEC com códigos em bloco

A questão que se coloca neste momento é: se disponho de um meio de transmissão cujo comportamento é modelado por um BEC(ϵ), como devo proceder para transmitir a informação gerada por uma fonte (considere que a fonte produz informação binária, ou seja, a saída da fonte u_i em um determinado instante de tempo i é modelada como uma v.a. $U \in \mathcal{S} = \{0, 1\}$). Um modelo de sistema de comunicações que atinge este objetivo é apresentado na Fig. 2.

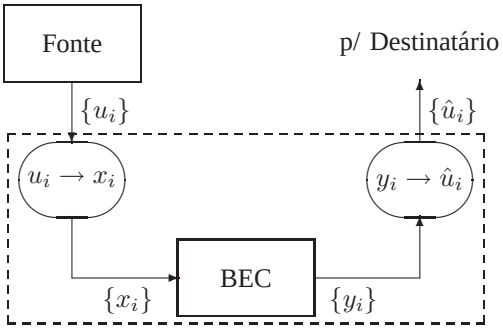

Fig. 2. Sistema de Comunicações com BEC(ϵ) (a interface $u_i \rightarrow x_i$ mapeia $u_i = 0$ em $x_i = a_1$ e $u_i = 1$ em $x_i = a_2$ e, de modo similar, a saída do canal, $y_i \in \mathcal{B}$, deve ser mapeada no símbolo a ser entregue ao destinatário $\hat{u}_i \in \mathcal{S}$, — pode-se usar, por exemplo, o mapa: $\hat{u}_i = 0$ se $y_i = b_1$ e $\hat{u}_i = 1$ caso contrário).

É fácil ver que, no sistema da Fig. 2, todos os apagamentos recebidos foram mapeados em "1" pela interface P_k/S_1 — que tudo se passa, portanto, como se a transmissão dos dados, produzidos pela fonte, ao destinatário, esteja sendo feita através de um canal Z (demarcado pela caixa com linha tracejada) cujo alfabeto de entrada é $\mathcal{A}$ e cujo alfabeto de saída é $\mathcal{B}_Z = \mathcal{A}$. Em resumo construiu-se um sistema, é fácil ver, que transformou o BEC (com entrada x e saída y, com probabilidade de apagamento $P[Y = b_2 | X = a_\ell] = \epsilon$ para $\ell \in \{1, 2\}$), em um canal Z (com entrada u e saída $\hat{u}$, com probabilidade de transição $P[\hat{U} = 1 | U = 0] = \epsilon$). [1]

Se a questão colocada fosse: dispõe-se de um meio de transmissão cujo comportamento é modelado por um BEC(ϵ), com probabilidade de apagamento ϵ (conforme caracterizado na Fig. 1), como devo proceder para transmitir ao destinatário, e entregar a informação gerada por uma fonte binária, sem erro? (Entenda-se "sem erro" como, usando a ideia de Shannon, "com probabilidade de erro tão pequena quanto se queira.") O diagrama em blocos de um sistema de comunicações que atinge este objetivo é apresentado na Fig. 3 — de acordo com o teorema da codificação para canal, a transmissão sem erro é teoricamente possível desde que a taxa R do codificador não ultrapasse a capacidade do canal.

Códigos como os códigos turbo e LDPC revolucionaram a prática de transmissão de informação através de canais não-ideais [8] (também denominados canais ruidosos, os canais não-ideais, que modelam a totalidade dos meios de transmissão encontrados na prática, têm a sequência de símbolos recebidos, isto é observados na saída do meio de transmissão, modelada por um vetor aleatório $\mathbf{Y}$ tal que $P[\mathbf{X} = \mathbf{x} | \mathbf{Y} = \mathbf{y}]$ é nula ou igual a um. Códigos desta família de códigos, com taxas muito próxima da capacidade do canal, podem ser encontrados, tornando possível construir sistemas de comunicações (como apresentado na Fig. 3) que operam com probabilidade de erro tão pequena quanto se queira. Tais códigos, com taxa fixa, permitem estabelecer sistemas que operam eficientemente se as condições do meio de transmissão não se alteram significativamente ao longo do período de transmissão — tais códigos, de taxa fixa, são inadequados se a capacidade, do canal que modela o meio de transmissão, cai abaixo da taxa do código (violando as condições do teorema da codificação para canal) ou ainda, se capacidade atinge valores muito maiores do que a taxa do código (tornando o sistema ineficiente). São inadequados também se o canal que modela o meio tem caracterização probabilística desconhecida. Para contornar estas dificuldades, técnicas denominadas de compatibilizadoras de taxa (*rate compatible* [1]) foram propostas. Mais recentemente alavancado pelo conceito de fontana digital [2], foram criados os códigos fontanais [3], [4] — estes são códigos cuja taxa pode ser reduzida progressivamente através do envio de uma sequência infindável de símbolos redundantes ao receptor e, por esta razão, vêm sendo denominados, na literatura inglesa, por *rateless codes*. Na Seção V os códigos LT, propostos por Luby [3] são apresentados.

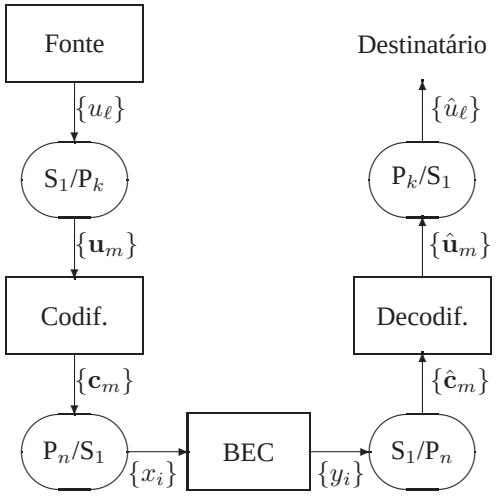

Fig. 3. Sistema de comunicações com código e um BEC: a interface S_1/P_k mapeia k símbolos binários da sequência $\{u_\ell\}$ em um vetor binário $\mathbf{u}_m$ com k componentes — e de forma idêntica operam as demais interfaces S/P e P/S; o codificador, de taxa $R = k/n$, mapeia o vetor $\mathbf{u}_m$, de dimensão k, em um vetor binário $\mathbf{c}_m$, de dimensão n.

IV. Transmissão através de BEC com códigos rateless

Os *códigos-LT* (*Luby-Transform Codes*) foram originalmente propostos para proteger dados a serem transmitidos através de um *Erasure Channel*. Por simplicidade considere que os dados serão transmitidos através de um meio modelado por um

[1] A capacidade de um BEC(ϵ), sabe-se, é $C_{\text{BEC}} = 1 - \epsilon$, maior portanto, para um dado valor de ϵ, do que a capacidade $C_{\text{Z}} = \log\left(1 + (1-\epsilon)\epsilon^{\epsilon/(1-\epsilon)}\right)$ de um canal Z(ϵ) (evidenciando que a redução de um BEC(ϵ) a um canal Z(ϵ), representa um sacrifício de capacidade.)

BEC (*Binary Erasure Channel*) — o BEC é muito utilizado para modelar matemáticamente muitos meios de transmissão de informação (a Internet, por exemplo).
Com eficiência e baixa complexidade os códigos-*LT* permitem codificar um bloco $\mathbf{u}_i$, com k símbolos gerados pela fonte, mapeando este bloco em uma palavra-código $\mathbf{c}_i$, de comprimento N e transmitir fidedignamente esta informação por um BEC com capacidade C (bits/uso-do-canal) — se k é suficientemente grande, o número N de bits a serem transmitidos é aproximadamente $\frac{k}{C}$, o menor número de bits requerido pelo teorema de Shannon para codificação de canal. Códigos-*LT* podem ser vistos como um código de bloco linear, ou seja, o codificador realiza o processamento

$$\mathbf{c}_i = \mathbf{u}_i \cdot \mathbf{G}.$$

Diferentemente dos códigos clássicos, cujo projeto se baseia na maximização da distância-mínima do código, os códigos-*LT* são projetados construindo-se uma distribuição de graus que minimiza a probabilidade de falha de decodificação.

V. CÓDIGOS LT

A descrição dos códigos-*LT* pode ser encontrada em vários artigos [3], [6], [8]. Para introduzir a notação a ser usada apresentamos uma descrição detalhada nesta seção.

Considere que um vetor de entrada $\mathbf{u} = (u_1, u_2, \ldots, u_k)$ será codificado. Ambos os vetores $\mathbf{u}$ e $\mathbf{c} = (c_1, c_2, \ldots, c_N)$ (saída do decodificador) possuem componentes que pertencem ao conjunto $\{0, 1\}$. As componentes c_j, $j = 1, \ldots, N$ relacionam-se ao vetor de entrada por

$$c_j = \mathbf{u} \cdot \mathbf{g}_j = \sum_{\ell=1}^{k} u_\ell g_{j,\ell}. \tag{2}$$

Em (2), $\mathbf{g}_j = (g_{j,1}, g_{j,2}, \ldots, g_{j,k})$, são vetores-coluna da matriz $\mathbf{G}$, geradora do código, de dimensões $k \times N$, cuja construção segue fundamentos teóricos conforme apresentados em [3] e extendidos em outros artigos [6]. Em resumo esta construção estabelece o peso de Hamming do vetores-coluna $\mathbf{g}_j$, $j \in \mathbb{N}$ e os elementos deste vetor que terão valor "1". Os pesos dos vetores-coluna, $\mathbf{d} = (d_1, d_2, \ldots, d_N)$, devem ser escolhidos aleatoriamente de acordo com uma distribuição de probabilidades e, neste sentido, são uma instância de uma sequência de v.a.'s $(D_1, D_2, \ldots, D_N)$, $j = 1, \ldots, N$, independentes e igualmente distribuídas com probabilidades

$$P(D_j = i) = p_i, \ (i = 1, \ldots, k; \ j = 1, \ldots, N). \tag{3}$$

Uma vez determinado o peso d_j do vetor-coluna $\mathbf{g}_j$ resta escolher quais componentes deste vetor serão igualadas a um. Esta escolha é realizada selecionando-se, aleatoriamente e com igual probabilidade, as d_j componentes de $\mathbf{u}$ a serem somadas (soma módulo-2) para produzir o símbolo c_j de saída do codificador (conforme equação (2)).

Neste ponto introduz-se uma correspondência entre o codificador descrito por (2) e um grafo bi-particionado em que cada símbolo de entrada u_i é associado ao nó de símbolo de informação α_i e cada símbolo da palavra-código c_j é associado a um nó de símbolo codificado β_j. Neste grafo a aresta e_{ij} conecta o nó-de-informação α_i ao nó-de-codificação β_j se o elemento $g_{ij} = 1$ da matriz-geradora é igual a "1". A Fig. 4 illustra o grafo $\mathcal{G}$ correspondente à matriz

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{4}$$

Observe que $\mathbf{d}$, o vetor de graus usado para construir esta matriz é $\mathbf{d} = (d_1, d_2, d_3, d_4, d_5, d_6) = (1, 3, 2, 1, 3, 2)$.

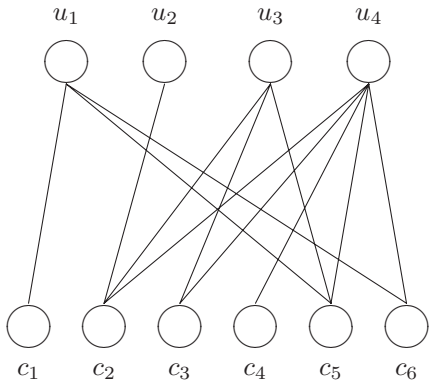

Fig. 4. Grafo $\mathcal{G}$ associado à matriz $\mathbf{G}$ em (4).

O algoritmo a seguir sumariza o processo de codificação (este codificador-*LT* com k símbolos de entrada e n símbolos de saída é dito operar com taxa fixa $R_{LT} = \frac{k}{n}$ ou, de forma similar, com taxa de expanção $\rho_{LT} = \frac{n}{k}$).

Algoritmo LT para Codificação

1) *Initializar o codificador:*
   a) Construir o vetor de graus $\mathbf{d} = (d_1, d_2, \ldots, d_n)$.
   b) Construir a matriz , $\mathbf{G_d}$, construindo as colunas $\mathbf{g}_j = (g_{j,1}, g_{j,2}, \ldots, g_{j,k})$ de acordo com o vetor $\mathbf{d}$.
   c) Faça a entrada do codificador igual a $\mathbf{u} = (u_1, u_2, \ldots, u_k)$.
2) *Gerar a palavra-código a ser transmitida:* encontrar a saída do codificador, $\mathbf{c} = (c_1, c_2, \ldots, c_n)$ de acordo com a equação (2).

Para a descrição do procedimento de decodificação considere que a palavra-código $\mathbf{c}$ foi transmitida e que o vetor $\mathbf{v} = (v_1, v_2, \ldots, v_n)$, foi recebido. Os símbolos $v_i = b_2$, uma vez identificados, não serão utilizados e, portanto, a primeira ação do procedimento de decodificação é expurgar estes símbolos — considera-se que os mecanismos de sincronização se encarregam de fornecer os índices i dos símbolos v_i recebidos e que o vetor de graus $\widetilde{\mathbf{d}}_1 = (\widetilde{d}_{1,1}, \widetilde{d}_{1,2}, \ldots, \widetilde{d}_{1,n'})$ e as arestas dos $n' \leq n$ símbolos $v_i \neq b_2$ (símbolos não-apagados) $\widetilde{\mathbf{c}}_1 = (\widetilde{c}_{1,1}, \widetilde{c}_{1,2}, \ldots, \widetilde{c}_{1,n'})$

são conhecidas pelo decodificador.

A tarefa de decodificação consiste em estimar a informação transmitida $\mathbf{u}$. Para iniciar, no Passo 1, o decodificador é inicializado com o vetor $\widetilde{\mathbf{c}}_1$, de dimensão n', e os respectivos graus associados; os símbolos $\widetilde{\mathbf{u}}_1 = (\widetilde{u}_{1,1}, \widetilde{u}_{1,2}, \ldots, \widetilde{u}_{1,k})$, são igualados a a_3 (símbolo desconhecido)i.e., $\widetilde{u}_{1,\ell} = a_3$ para todo $\ell = 1, \ldots, k$, ou seja, os símbolos estimados são declarados desconhecidos.

O procedimento de decodificação inicia com o vetor $\left(\widetilde{\mathbf{c}}_1, \widetilde{\mathbf{d}}_1, \widetilde{\mathbf{u}}_1\right)$ e produz a sequência de vetores $\{\left(\widetilde{\mathbf{c}}_1, \widetilde{\mathbf{d}}_1, \widetilde{\mathbf{u}}_1\right), \left(\widetilde{\mathbf{c}}_2, \widetilde{\mathbf{d}}_2, \widetilde{\mathbf{u}}_2\right), \ldots, \left(\widetilde{\mathbf{c}}_m, \widetilde{\mathbf{d}}_m, \widetilde{\mathbf{u}}_m\right)\}$ através da transformação sucessiva do trio de vetores $\left(\widetilde{\mathbf{c}}_m, \widetilde{\mathbf{d}}_m, \widetilde{\mathbf{u}}_m\right)$ e matriz associada $\mathbf{G}_{\widetilde{\mathbf{d}}_m}$ no trio de vetores $\left(\widetilde{\mathbf{c}}_{m+1}, \widetilde{\mathbf{d}}_{m+1}, \widetilde{\mathbf{u}}_{m+1}\right)$ com matriz associada $\mathbf{G}_{\widetilde{\mathbf{d}}_{m+1}}$. Para definir esta transformação, que será referida como *redução de grafo* atente para o fato: a cada trio $\left(\widetilde{\mathbf{c}}_m, \widetilde{\mathbf{d}}_m, \widetilde{\mathbf{u}}_m\right)$, existe um grafo $\mathcal{G}_m$ univocamente associado ao trio.

Seja $\mathcal{G}_m = (\mathcal{A}_m, \mathcal{B}_m, \mathcal{E}_m)$ o grafo associado a $\left(\widetilde{\mathbf{c}}_m, \widetilde{\mathbf{d}}_m, \widetilde{\mathbf{u}}_m\right)$. O conjunto $\mathcal{B}_m = \{\beta_{m,j_1}, \beta_{m,j_2}, \ldots, \beta_{m,J_m}\}$, de cardinalidade J_m, é o conjunto de vértices associados ao vetor $\widetilde{\mathbf{c}}_m$ — estes serão chamados de nós$-\beta$ ou vértices$-\beta$. O conjunto de nós$-\alpha$ $\mathcal{A}_m = \{\alpha_{m,i_1}, \alpha_{m,i_2}, \ldots, \alpha_{m,I_m}\}$, de cardinalidade I_m, é o conjunto de vértices associados ao vetor $\widetilde{\mathbf{u}}_m$. $\mathcal{E}_m$, de cardinalidade $\sum_{x=1}^{J_m} \widetilde{d}_{m,x}$, é o conjunto de arestas $(\beta_{m,j}, \alpha_{m,i})$ onde (j,i) são as posições na matriz $\mathbf{G}_{\widetilde{\mathbf{d}}_m}$ que correspondem a $g_{m,j,i} = 1$.

Definição (Redução de Grafo) Diz-se que $\mathcal{G}_m$ é um grafo redutível se há um componente do vetor $\widetilde{\mathbf{d}}_m$, igual a 1. A redução deste gráfico é obtida identificando-se o correspondente vértice $\beta_{m,j'}$ de grau 1 com valor associado igual a $\widetilde{c}_{m,j'}$, w de menor índice j'. Seja $(\beta_{m,j'}, \alpha_{m,i'})$ a aresta conectada a este nó. A redução prossegue construindos-e o vetor $\widetilde{\mathbf{u}}_{m+1}$ com os mesmos valores atribuídos ao vetors $\widetilde{\mathbf{u}}_m$ exceto pelo valor atribuído ao vértice $\widetilde{u}_{m+1,i'}$ (valor a_3, desconhecido) que é substituído pelo valor $\widehat{c}_{m,j'}$. O subconjunto de nós$-\beta$ que são vizinhos de $\alpha_{m,i'}$, o nó-α cujo valor foi recém atualizado, ou seja, o conjunto $\mathcal{N}_{m,i'} = \{\beta_{m,j'_1}, \beta_{m,j'_2}, \ldots, \beta_{m,J'_m}\}$, é a seguir identificado e o vetor $\widetilde{\mathbf{c}}_{m+1}$ é criado — este vetor possui os mesmas coordenadas do vetor $\widetilde{\mathbf{c}}_m$ excetopelas componentes $\{c_{m,j'_1}, c_{m,j'_2}, \ldots, c_{m,J'_m}\}$ que são substituídas por $\{c_{m,j'_1} \oplus \widehat{c}_{m,j'}, c_{m,j'_2} \oplus \widehat{c}_{m,j'}, \ldots, c_{m,J'_m} \oplus \widehat{c}_{m,j'}\}$. Finalmente chega-se ao grafo reduzido $\mathcal{G}_{m+1}$, grafo obtido quando se elimina os vertices$-\alpha$, $\alpha_{m,i'}$ e os vertices$-\beta$ pertencentes a $\mathcal{N}_{m,i'}$ (e, naturalmente, as aretsas correspondentes). $\diamondsuit$

O grafo que não é redutível é dito irredutível. Para ilustrar estas ideias considere que $\mathcal{G}_1 = \mathcal{G}$ é o grafo mostrado na Fig. 4. O grafo $\mathcal{G}_2$ apresentado na Fig. 5 é uma redução de $\mathcal{G}_1$.

O procedimento de decodificação é sumarizado a seguir.

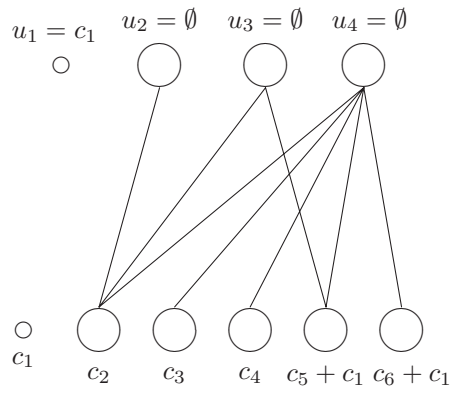

Fig. 5. Grafo $\mathcal{G}_2$, obtido pela reduçao de $\mathcal{G}_1$.

Algoritmo LT para Decodificação

1) *Inicializar o passo recursivo da decodificação:*
   a) Inicializar o contador de passos recursivos: faça $m = 1$;
   b) Inicializar o vetor de graus: faça $\widetilde{\mathbf{d}}_m = (\widetilde{d}_{1,1}, \widetilde{d}_{1,2}, \ldots, \widetilde{d}_{1,n'})$;
   c) Especificar a matriz $\mathbf{G}_{\widetilde{\mathbf{d}}_1}$: construa, levando em conta o vetor de grau $\widetilde{\mathbf{d}}_1$), as colunas $\mathbf{g}_{m,j} = (g_{1,j,1}, g_{1,j,2}, \ldots, g_{1,j,k})$, da matriz;
   d) Inicializar o vetor de entrada do decodificador : faça $\widetilde{\mathbf{c}}_m = (\widetilde{c}_{1,1}, \widetilde{c}_{1,2}, \ldots, \widetilde{c}_{1,n'})$;
   e) Inicializar o vetor de saída do decodificador : faça $\widetilde{\mathbf{u}}_m = (\widetilde{u}_{1,1}, \widetilde{u}_{1,2}, \ldots, \widetilde{u}_{1,k})$ onde $\widetilde{u}_{1,\ell} = a_3$ para todo $\ell = 1, 2, \ldots, k$;
   f) Inicializar o grafo-do-decodificador: construa o grafo $\mathcal{G}_1 = (\mathcal{A}_m, \mathcal{B}_m, \mathcal{E}_m)$ onde, $\mathcal{A}_m = \{\alpha_{1,i_1}, \alpha_{1,i_2}, \ldots, \alpha_{1,I_1}\}$, de cardinalidade I_1, é o conjunto de vértices associado ao vetor $\widetilde{\mathbf{u}}_1$; $\mathcal{B}_m = \{\beta_{1,j_1}, \beta_{1,j_2}, \ldots, \beta_{1,J_1}\}$, de cardinalidade J_1, é o conjunto de vértices associado ao vetor $\widetilde{\mathbf{c}}_1$ e; $\mathcal{E}_1$, de cardinalidade $\sum_{x=1}^{j_1} \widetilde{d}_{1,x}$, é o conjunto de arestas $(\beta_{1,j}, \alpha_{1,i})$ onde (j,i) são as posições na matriz $\mathbf{G}_{\mathbf{d}_1}$ que correspondem a $g_{1,j,i} = 1$.
2) *Testar redutibilidade do grafo:* Se o grafo $\mathcal{G}_m$ é irredutível, declare que houve falha de decodificação e siga para o passo final (Passo 5);
3) *Fazer redução de grafo:* $\mathcal{G}_{m+1} = (\mathcal{A}_{m+1}, \mathcal{B}_{m+1}, \mathcal{E}_{m+1})$ (observe que $\mathcal{G}_{m+1}$ é o grafo reduzido associado a $\left(\widetilde{\mathbf{d}}_{m+1}, \widetilde{\mathbf{c}}_{m+1}, \widetilde{\mathbf{u}}_{m+1}\right)$);
4) *Iterar:* Se existe um símbolo $\widetilde{u}_{m+1,\ell} = a_3$ para algum $\ell = 1, 2 \ldots, k$, incremente o contador de recursão ($m = m + 1$) e retorne ao Passo 2;
5) *Parar:* a sequência transmitida estimada é $\widehat{\mathbf{u}} = \widetilde{\mathbf{u}}_{m+1}$.

VI. CÓDIGOS LT E LT-ENCURTADO

Códigos-*LT* encurtados (*sLT*) derivados de códigos-*LT* sistemáticos [6] serão a seguir descritos.

Um bloco $\mathbf{u} = (u_1, u_2, \ldots, u_k)$, conhecido por ambos o codificador e decodificador, é inicialmente concatenado com

um bloco $\mathbf{s} = (s_1, s_2, \ldots, s_r)$. O vetor resultante $\mathbf{u}' = \big((s_1, s_2, \ldots, s_r) \circ (u_1, u_2, \ldots, u_k)\big)$ tem comprimento $k' = r + k$. Este bloco atravessa um codificador de taxa $R' = \frac{k'}{N}$ que produz em sua saída o vetor

$$\begin{aligned} \mathbf{c}' &= (c'_1, c'_2, \ldots, c'_N) \\ &= (s_1, \ldots, s_r, u_1, \ldots, u_k, c'_{k'+1}, \ldots, c'_N). \end{aligned} \quad (5)$$

A saída do codificador-*sLT* (codificador encurtado), que será efetivamente transmitida é

$$\begin{aligned} \mathbf{c} &= (c_1, c_2, \ldots, c_n) \\ &= (u_1, \ldots, u_k, c'_{k'+1}, \ldots, c'_N), \end{aligned} \quad (6)$$

onde $n = N - r$. Ou seja, transmite-se a parte do vetor $\mathbf{c}'$ que resta quando o vetor $\mathbf{s}$ é suprimido.

O decodificador-*sLT* é semelhante a um decodificador-*sLT* com a palavra estimada inicial

$$\widetilde{\mathbf{u}}_1 = (\widetilde{u}_{1,1}, \widetilde{u}_{1,2}, \ldots, \widetilde{u}_{1,k'}) \quad (7)$$

com $\widetilde{u}_{1,\ell} = s_\ell$ for all $\ell = 1, 2, \ldots, r$, $\widetilde{u}_{1,\ell} = E$ for all $\ell = r+1, r+2, \ldots, k'$ e, $\widetilde{\mathbf{c}}_1$, o vetor da primeira iteração (de comprimento $N' = n' + r$), inicializado por

$$\begin{aligned} \widetilde{\mathbf{c}}_1 &= (\widetilde{c}_{1,1}, \widetilde{c}_{1,2}, \ldots, \widetilde{c}_{1,N'}) \\ &= (s_1, \ldots, s_r, \widetilde{u}_{1,1}, \ldots, \widetilde{u}_{1,k}, c'_{k'+1}, \ldots, c'_N). \end{aligned} \quad (8)$$

Se não há falha de decodificação, na saída $\widehat{\mathbf{u}} = \widetilde{\mathbf{u}}_m = \mathbf{u}$, caso contrário o decodificador-*sLT* repassará ao destinatário, no passo final de decodificação, a informação $\widehat{\mathbf{u}} = \widetilde{\mathbf{u}}_m \neq \mathbf{u}$.

Observe que o fator de expansão do codificador-*sLT* é $\rho_{sLT} = \frac{n}{k}$, enquanto o codificador mãe, codificador-*LT* que mapeia $\mathbf{u}'$ em $\mathbf{c}'$, possui fator de expansão $\rho' = \frac{n+r}{k+r}$.

Resultados experimentais, comparando o desempenho de sistemas com códigos *LT* o desempenho de sistemas com códigos *sLT* evidenciam que os segundos apresentam melhor desempenha. O gráfico da Fig. 6 exibe a probabilidade de falha de decodificação dos sistemas que utilizam códigos LT e que utilizam códigos sLT. A transmissão de blocos de valores pequenos apenas ($k = 500$ e $k = 1000$) apenas foram simulados. Conjetura-se que a diferença de desempenho seja mais modesta quando o tamanho do bloco cresce. Os resultados mostraram uma melhoria de desempenho dos sistemas com códigos sLT quando medido em termos da probabilidade de erro (nos casos em que falha de decodificação foi observada). O resultados foram obtidos com códigos de Luby sistemáticos. Simulações quando informação conhecida é utilizada com códigos não-sistemáticos assim como simulações com blocos de comprimentos mais elevados estão em curso.

VII. Conclusão

Uma prática comum quando se procura construir códigos clássicos é obter um novo código a partir de um código conhecido efetuando modificações tais como puncionamento, extensão, etc. Aparentemente estas técnicas são desinteressantes quando se trata de códigos fontanais

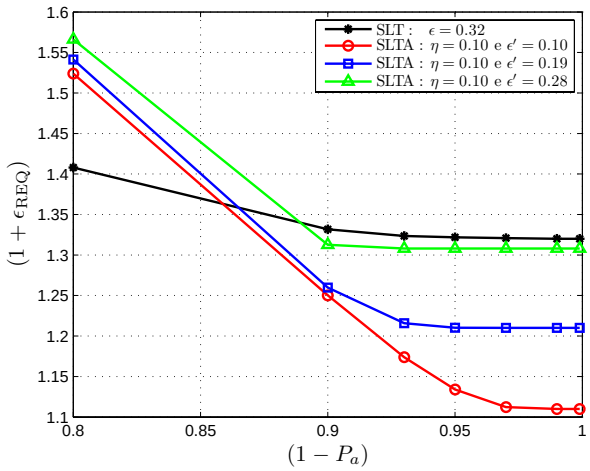

Fig. 6. Fator de expansão $(1 + \epsilon_{REQ})$ versus Qualidade do Canal (BEC), para diversos valores do encurtamento r. O sistema que usa código LT com $k = 500$, $n = 555$, $\rho_{LT} = 1.11$ é comparado com o sistema que usa código sLT tal que $\rho_{sLT} = 1.11$.

(por si mesmo, compatibilizadores de taxa). Resultados obtidos através de simulação usando blocos de comprimento relativamente curtos (blocos de nformação com $k = 500$ e $k = 1000$), em estudos recentes mostraram, no entanto, que o desempenho do códigos fontanais com algumas destas modificações (examinamos os códigos de Luby encurtados) pode melhorar. Conjeturo que tal melhoria se deve à diluição da perda de informação (introduzida pelos apagamentos) é devida à informação compartilhada (conhecida por ambos o codificador e o decodificador) — note que o apagamento de um símbolo, de grau d, conectado a ℓ símbolos de entrada conhecidos , dilui a perda pois representa perder apenas $(d - \ell)/d$ da informação. Simulações com outros tipos de modificações do código-mãe e com blocos de comprimentos longos, est ao em curso, assim como estudos para emoldurar e validar teoricamente os resultados apresentados. Outros resultados podem ser encontrados na literatura [6]–[8].

Referências

[1] J. Haguenauer, *Rate compatible punctured convolutional codes (RCPC codes) and their applications*, IEEE Trans. Commun., vol. 36, no. 4, pp.389-400, Apr. 1988.

[2] J. Byers, M. G. Luby, M. Mitzenmacher, and A. Rege, *A digital fountain approach to reliable distribuition of bulk data*, in Proceedings of ACM SIGCOMM, Vancouver, Canada, Sept. 1998.

[3] M. G. Luby, *LT-codes*, in Proceedings of 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 271-280, Nov. 16-19, Vancouver, Canada, 2002.

[4] M. A. Shokrollahi, *Raptor Codes*, IEEE Trans. Inf. Theory, vol. 52, pp.2551-2567, June 2006. IEEE Symposium Inf. Theory, p.36, 2004.

[5] R. E. Blahut, *Algebraic Codes for Data Transmission*, Cambridge, England: Cambridge University Press, 2003.

[6] T. Nguyen, L. Yang and L. Hanzo *Systematic Luby Transform codes and their soft decoding*, in IEEE Workshop on Signal Processing Systems (SiPS), Shanghai, China, October 2007.

[7] T. L. Grobler et alli *Systematic Luby Transform codes as Incremental Redundancy scheme*, IEEE Africon, Livingstone, Zambia, Sept. 2011.

[8] N. Bonello, S. Chen, and L. Hanzo *Low-density parity-check codes and their rateless relatives.*, IEEE Commun. Surveys Tuts., vol. 13, no. 1, pp. 03-26, 2011.