Referências Bibliográficas

[1] 3GPP TSG-SA WG4 S4-AHP238. **Specification Text for Systematic Raptor Forward Error Correction**. PSM SWG, Sophia Antipolis, France, Abril 2005.

[2] BYRES, J. W.; LUBY, M.; MITZENMACHER, M. ; REGE, A.. **A Digital Fountain Approach to Reliable Distribution of Bulk Data**. Proceedings of ACM SIGCOMM '98, p. 56–57, Setembro 1998.

[3] CATALDI, P.; SHATARSKI, M.; GRANGETTO, M. ; MAGLI, E.. **Implementation and performance evaluation of LT and Raptor codes for multimedia applications**. Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, p. 263–266, Dezembro 2006.

[4] ELIAS, P.. **Coding for two noisy channels**. Information Theory, Third London Symposium, p. 61–76, 1955.

[5] BELTRÃO NETO, HUMBERTO. **Estudo de uma classe de códigos sem-taxa para transmissão de dados em canais com apagamento**. Dissertação de Mestrado, Universidad Federal de Pernambuco, 2007.

[6] LUBY, M.. **LT codes**. Proc. of the 43rd Annual IEEE Symp. on Foundation of Comp. Sc., p. 271–280, Novembro 2002.

[7] MA, Y.; YUAN, D. ; ZHANG, H.. **Fountain Codes and Applications to Reliable Wireless Broadcast System**. Proceedings of 2006 IEEE Information Theory Workshop, 2006.

[8] MACKAY, D. J. C.. **Information Theory, Inference, and Learning Algorithms**. Cambridge University Press, 2003.

[9] MAYMOUNKOV, P.; MAZIERES, D.. **Rateless codes and big downloads**. In: Proc. of the 2nd International Workshop Peer-to-Peer System, 2003.

[10] MITZENMACHER, M.. **Digital Fountains: A Survey and Look Forward**. IEEE Information Theory Workshop, p. 24–29, Outubro 2004.

[11] NGUYEN, T.; YANG, L. ; HANZO, L.. **Systematic Luby Transform Codes And Their Soft Decoding**. 2007 IEEE Workshop on Signal Processing Systems in Shanghai, p. 67–72, Outubro 2007.

[12] NGUYEN, T.; YANG, L. ; HANZO, L.. **An Optimal Degree Distribution Design And A Conditional Random Integer Generator For The Systematic Luby Transform Coded Wireless**. WCNC, 2008.

[13] MAYMOUNKOV, P.. **Online codes**. Technical report, NYW, Relatório Técnico 2003-833, Novembro 2002.

[14] LUBY, M.. **Information additive code generator and decoder for communication systems**. U.S. Patent No. 7,233,264 B2, Junho 19, 2007.

[15] PUDUCHERI, S.; KLIEWER, J. ; FUJA, T.. **The Design and Performance of Distributed LT codes**. IEEE Transactions on Information Theory, Vol. 53, No 10:pag. 3740–3754, Outubro 2007.

[16] REED, I. S.; SOLOMON, G.. **Polynomial Codes Over Certain Finite Fields**. J. Soc. Indust. Appl. Math, Vol. 8:pag. 300–304, 1960.

[17] RICHARDSON, T. J.; URBANKE, R.. **The capacity of low-density parity-check codes under message-passing decoding**. IEEE Trans. Information Theory, vol. 47:pag. 599–618, Fevereiro 2001.

[18] SAMAD, W. A.. **Efficient Video on Demand (VoD) Services in Broadcast Environments**. Dissertação de Mestrado, School of Electrical Engineering, Stockholm - Sweden, Abril 2006.

[19] SASAKI, C.; HASEGAWA, T. ; KOBAYASHI, S.. **On Unicast based Recovery for Multicast Content Distribution considering XOR-FEC**. Asia-Pacific Conference on Communications, Outubro 2005.

[20] SHANNON, C.. **A mathematical theory of communications**, volumen 27, pag. 379-423 e 623-656. The Bell System Technical Journal, Julho e Outubro 2004.

[21] SHOKROLLAHI, A.. **Raptor codes**. IEEE Transactions on Information Theory, Vol. 52, No 6:pag. 2551–2567, Junho 2006.

[22] SHOKROLLAHI, A.; LUBY, M.. **Systematic Encoding and Decoding of Chain Reaction Codes**. U.S. Patent No. 6,909,383, Junho 21, 2005.

[23] TANNER, R. M.. **A recursive approach to low-complexity codes**. IEEE Trans. Information Theory, p. 533–547, Setembro 1981.

[24] TEE, R.; NGUYEN, T.; YANG, L. ; HANZO, L.. **Serially Concatenated Luby Transform Coding And Bit-Interleaved Coded Modulation Using Iterative Decoding For The Wireless Internet**. Proceedings of VTC 2006 Spring, Melbourne, vol. 138:pag. 177–182, Maio 2006.

[25] TIRROMEN, T.. **Optimizing the degree distribution of LT codes**. Dissertação de Mestrado, HELSINKI UNIVERSITY OF TECHNOLOGY, Março 2006.

[26] BYRES, J. W.; LUBY, M.; MITZENMACHER, M. ; REGE, A.. **A Digital Fountain Approach to Asynchronous Reliable Multicast**. IEEE J. on Selected Areas in Communications, Special Issue on Network Support for Multicast Communications, Vol. 20:pag. 1528–1540, Agosto 2002.

[27] LUBY, M.; MITZENMACHER, M. ; SHOKROLLAHI, A.. **Efficient Erasure Correction Codes**. IEEE Trans. on Information Theory, Vol. 47:pag. 569–584, Fevereiro 2001.

[28] LUBY, M.; WATSON, M.; GASIBA, T.; STOCKHAMMER, T. ; XU, W.. **Raptor Codes for Reliable Download Delivery in Wireless Broadcast Systems**. Conference: CCNC' 2006, Julho 2005.

[29] WICKER, S. B.. **Error Control Systems for Digital Communication and Storage**. Prentice Hall, Upper Saddle River, NJ07458 - USA, 1995.

A
Prova do Teorema 3.1

Usamos prova por indução para $d \in \{2, \ldots, k\}$:

1. Base da indução: com $d = 2$ o resultado provinde de (3-4), isto é,

$$h_t(2) = (k - t)/2$$

2. Etapa de indução: nossa hipótese de indução é que o Teorema 3.1 é verdadeiro para n, ou seja,

$$h_t(n) = \frac{k-t}{n(n-1)}$$

Agora vamos provar o caso de $n + 1$ utilizando (3-6):

$$\begin{aligned}
h_t(n+1) &= \frac{k-t}{n+1}\left(h_{t+1}(n) - h_t(n)\right) + \frac{n}{n+1}h_t(n) \\
&= \frac{k-t}{n+1}\left(\frac{k-t-1}{n(n-1)} - \frac{k-t}{n(n-1)}\right) + \frac{n}{n+1}.\frac{k-t}{n(n-1)} \\
&= -\frac{k-t}{(n+1)n(n-1)} + \frac{n(k-t)}{(n+1)n(n-1)} \\
&= \frac{(n-1)(k-t)}{(n+1)n(n+1)} \\
&= \frac{k-t}{n(n+1)}.
\end{aligned}$$

o que é igual ao descrito no Teorema 3.1.

Pela indução da propriedade de números naturais, o Teorema 3.1 é verdadeiro para todo $d \in \{2, \ldots, k\}$.

B
Códigos Raptor

Códigos Raptor foram introduzidos por Shokrollahi em 2003 [21], como uma extensão dos códigos LT. Os códigos Raptor são uma classe de códigos universais, no sentido que para um determinado número de símbolos de entrada k, e qualquer overhead $\varepsilon_{Raptor} > 0$, o código Raptor pode gerar um número de símbolos codificados potencialmente ilimitado tal que qualquer subconjunto de $k(1+\varepsilon_{Raptor})$ símbolos codificados é suficiente para recuperar os k símbolos de entrada originais com alta probabilidade. Cada símbolo codificado é gerado utilizando $O\left(-\log \varepsilon_{Raptor}\right)$ operações, e os símbolos de entrada originais são recuperados a partir dos $k(1+\varepsilon_{Raptor})$ símbolos com $O(-k \log \varepsilon_{Raptor})$ operações.

Uma versão totalmente especificada dos códigos Raptor foi recentemente aprovada em [1], como um meio para difundir dados de maneira eficiente através de uma rede *broadcast*. Os códigos Raptor podem ser abordados por diferentes ângulos. Por um lado, eles podem ser vistos como um código de bloco sistemático especificado por uma matriz geradora, por outro lado a idéia de códigos fontanais aleatórios também é inerente. Um código Raptor, conforme especificado por MBMS (*Multimedia Broadcast/Multicast Services*) é um código fontanal sistemático gerando n símbolos codificados.

Nas seções seguintes, explicamos com detalhes a estrutura, o processo de codificação e decodificação do código Raptor. Esta descrição e simbologia utilizada seguem de modo muito próximo as abordagens apresentadas em [18, 28].

B.1
Códigos Raptor não-sistemáticos

Os códigos Raptor [21] podem ser codificados e decodificados em tempo que varia linearmente com k e conseguir um desempenho superior aos códigos LT. A novidade é a introdução de uma etapa de pré-codificação externa aplicando um código de bloco sistemático de comprimento fixo, por meio da qual a parte não sistemática da matriz geradora é denotada por uma matriz $\mathbf{G}_P$ de

dimensão $(m \times k)$. Este código gera $L = k + m$ símbolos pre-codificados com m denotando o número de símbolos de paridade, como mostrado na Figura B.1.

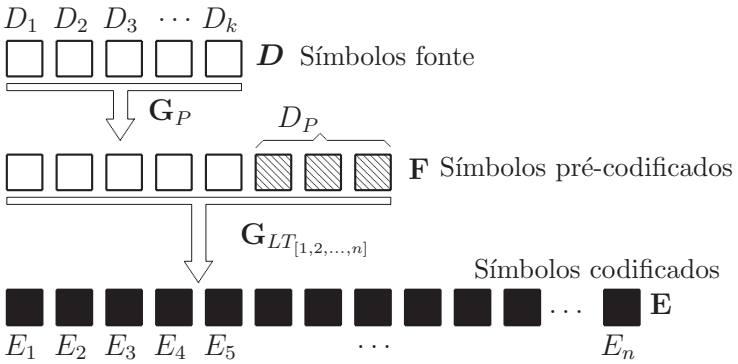

Figura B.1: Etapa de Pré-codificação em códigos Raptor.

Seja $\mathbf{D}$ o vetor de dimensão $(k \times 1)$, que contém os k símbolos de entrada que entram no código Raptor não-sistemático. Então os símbolos de paridade pre-codificados denotados pelo vetor $\mathbf{D}_P$ são obtidos como

$$\mathbf{D}_P = \mathbf{G}_P \cdot \mathbf{D}. \tag{B-1}$$

O conjunto de símbolos pre-codificados, denotado pelo vetor $\mathbf{F}$ de dimensão $(k+m)\times 1$, é então dado por $\left(\mathbf{D}^T\ \mathbf{D}_P^T\right)^T$. Estes símbolos pré-codificados servem agora como símbolos de entrada para o código LT interno subseqüente que opera da mesma maneira como um código LT convencional e mantém a propriedade de produzir n símbolos codificados Raptor. Notar, entretanto, que a distribuição de graus é mudada para os códigos Raptor. Os símbolos codificados denotados pelo vetor $\mathbf{E}_{[1:n]}$ de dimensão $(n \times 1)$, são obtidos por

$$\mathbf{E}_{[1:n]} = \mathbf{G}_{LT_{[1,2,\cdots,n]}} \cdot \left(\mathbf{D}^T\ \mathbf{D}_P^T\right)^T, \tag{B-2}$$

onde a matriz $\mathbf{G}_{LT_{[1,2,\cdots,n]}}$ de dimensão $n \times (k + m)$ é a matriz geradora do código LT. As Equações (B-1) e (B-2) podem ser representadas como

$$\mathbf{A}_{[1,2,\cdots,n]} \cdot \begin{pmatrix} \mathbf{D} \\ \mathbf{D}_P \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{E}_{[1:n]} \end{pmatrix}, \tag{B-3}$$

através do qual, a matriz $\mathbf{A}$ de dimensão $(m + n) \times (k + m)$ conhecida como "*matriz de codificação*" é definida por

$$\mathbf{A}_{[1,2,\cdots,n]} \triangleq \begin{pmatrix} \mathbf{G}_P\ \mathbf{I}_m \\ \mathbf{G}_{LT_{[1,2,\cdots,n]}} \end{pmatrix}, \tag{B-4}$$

e $\mathbf{I}_m$ é a matriz identidade $(m \times m)$. Notar que a matriz de codificação $\mathbf{A}$ não representa uma matriz geradora, mas fornece um conjunto de restrições tanto

para a etapa de pre-codificação quanto para o código LT interno. Do ponto de vista da codificação, o codificador Raptor não-sistemático apenas realiza a etapa adicional de calcular os símbolos pré-codificados. Então, um conjunto de n símbolos codificados consecutivos $\mathbf{E}_{[1:n]}$ é obtido de acordo com (B-2).

Mais uma vez, assume-se que no decodificador somente um subconjunto de símbolos codificados esta disponível, indicado pelo vetor identificador de símbolos codificados $\boldsymbol{i} = (i_1,\ i_2, \ldots,\ i_r)$. A decodificação de códigos Raptor é realizada baseada na matriz de decodificação $\mathbf{A}_{[i_1,i_2,\cdots,i_r]}$ definida como

$$\mathbf{A}_{[i_1,i_2,\cdots,i_r]} \triangleq \begin{pmatrix} \mathbf{G}_P\ \mathbf{I}_m \\ \mathbf{G}_{LT_{[i_1,i_2,\ldots,i_r]}} \end{pmatrix}. \tag{B-5}$$

Similar ao decodificador LT, o processo de decodificação para códigos Raptor consiste em resolver um sistema de equações lineares, especificamente

$$\mathbf{A}_{[i_1,i_2,\cdots,i_r]} \cdot \mathbf{F} = \left(\mathbf{0}^T\ E_{i_1}^T\ E_{i_2}^T\ \cdots\ E_{i_r}^T\right)^T, \tag{B-6}$$

onde $\mathbf{0}$ é um vetor nulo de dimensão $(m \times 1)$, e fazendo que os primeiros k símbolos pré-codificados correspondam aos k símbolos fonte

$$\mathbf{F}_{[1:k]} = \mathbf{D}. \tag{B-7}$$

É importante que a pré-codificação externa, assim como a distribuição de graus do código Raptor seja selecionada apropriadamente para obter um bom desempenho do código. Na realidade, para um código Raptor como especificado em MBMS [1], a etapa de pré-codificação consiste de dois códigos concatenados em série, através do qual os pre-códigos externo e interno representam códigos tais como LDPC com a estrutura da matriz geradora quase-regular (número constante de 1s por linha e por coluna).

Além disso, é interessante notar que a matriz de decodificação $\mathbf{A}_{[i_1,i_2,\cdots,i_r]}$ é uma generalização da matriz de codificação como definida em (B-4) porque os indices não são mais consecutivos, mas somente os indices recebidos são tomados em conta. Portanto, nos referimos no restante a qualquer matriz incluindo as restrições do código de um vetor arbitrário $\boldsymbol{i}$ como *"code constraint matrix"* e denotada como $\mathbf{A}_{\boldsymbol{i}}$.

B.2 Codificação Raptor sistemática.

Apesar de que o código Raptor não-sistemático tem um excelente desempenho, é sabido que para muitas aplicações o acesso direto aos dados originais é

benéfico, permitindo aos receptores que não são capazes de explorar os pacotes de redundância, participar ainda em uma sessão somente usando os pacotes de dados que contem os dados originais. Códigos sistemáticos podem proporcionar esta propriedade já que todos os símbolos fonte aparecem nos símbolos codificados. No entanto, como o código LT é um código não-sistemático devido a sua construção e o código interno do código Raptor é também um código LT, ambos códigos são inicialmente não-sistemáticos. No seguinte, mostramos brevemente como códigos Raptor não-sistemáticos podem ser transformados em um código sistemático [21, 22].

Como já foi mencionado, qualquer código fontanal pode ser visto como código em bloco linear regular representado por uma matriz geradora. É também conhecido da teoria de codificação, que as propriedades de um código são determinadas pelas palavras-código e não pela regra de codificação, isto é, o mapeamento de palavras de informação às palavras-código. Conseqüentemente, para obter um código sistemático, é necessário encontrar um mapeamento apropriado das palavras de informação às palavras-código tais que os primeiros k símbolos da palavra-código E_i correspondam aos primeiros símbolos fonte C_i, isto é

$$E_i = C_i \quad \forall\ i = 1, \ldots, k \tag{B-8}$$

onde $\mathbf{C}$ representa o vetor símbolo fonte, e C_i o i-ésimo símbolo fonte. Assumir que a matriz geradora de um código Raptor não-sistemático para os primeiros k símbolos é denotado pela matriz $\mathbf{G}_k$ de dimensão $(k \times k)$ tal que $\mathbf{E}_{[1:k]} = \mathbf{G}_k \cdot \mathbf{D}$. Então, para garantir (B-8) é suficiente multiplicar $\mathbf{C}$ pela inversa de $\mathbf{G}_k$ denotada como $\mathbf{G}_k^{(-1)}$ tal que a matriz geradora total do código sistemático proporciona a matriz identidade para as k primeiras posições. É obvio, que esta propriedade pode ser satisfeita se $\mathbf{G}_k$ tem inversa, qual é equivalente ao caso que a matriz tem posto completo (*full rank*) k.

A Figura B.2 mostra a estrutura de um código Raptor sistemático, qual pode ser representada por uma concatenação serial de $\mathbf{G}_k^{(-1)}$, o pré-código $\mathbf{G}_P$, e a matriz geradora $\mathbf{G}_{LT}$ do código LT . Aproveitando o fato que $\mathbf{E}_{[1:k]} = \mathbf{C}$, isto é, a Equação (B-8), e as Equações (B-1) e (B-2), obtemos

$$\mathbf{G}_k = \mathbf{G}_{LT_{[1,\ldots,k]}} \cdot \begin{pmatrix} \mathbf{I}_k \\ \mathbf{G}_P \end{pmatrix}. \tag{B-9}$$

Uma vez calculada esta matriz, os símbolos fonte podem ser facilmente mapeados em símbolos intermediários $\mathbf{D}$ como $\mathbf{D} = \mathbf{G}_k^{(-1)} \cdot \mathbf{C}$.

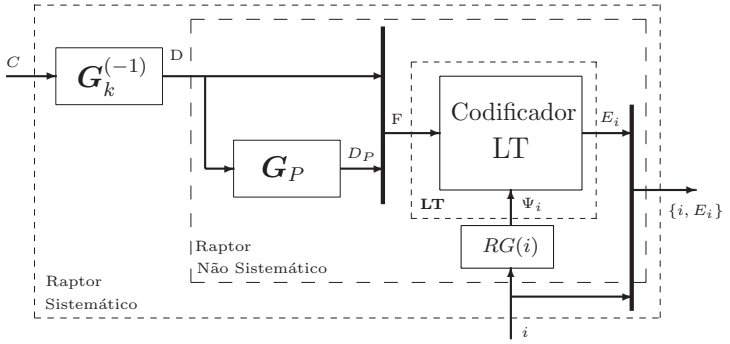

Figura B.2: Diagrama em bloco da codificação Raptor sistemática.

B.3
Construção de codificador e decodificador Raptor sistemático

Uma implementação direta seria construir uma matriz $\mathbf{G}_k$ invertível e aplicar a codificação Raptor não-sistemática como mostrado na Figura B.2. No entanto, este processo de construir $\mathbf{G}_k$ e calcular sua inversa $\mathbf{G}_k^{(-1)}$, pode ser muito ineficiente do ponto de vista computacional, então um procedimento de codificação diferente foi proposto em [1]. É sugerido que a codificação pode ser feita usando uma *"code constraint matrix"* específica, ou seja, a matriz de codificação conforme definida em (B-4) para $n = k$, isto é, $\mathbf{A}_{[1,\cdots,n=k]}$. Desta forma, os símbolos intermediários $\mathbf{D}$ não necessitam ser calculados explicitamente, e os símbolos pré-codificados $\mathbf{F}$ são obtidos logo por

$$\mathbf{A}_{[1,\cdots,k]} \cdot \mathbf{F} \equiv \left(\mathbf{0}^T \ \mathbf{C}^T\right)^T. \tag{B-10}$$

Notar que os símbolos intermediários $\mathbf{D}$ são obtidos inerentemente a partir de $\mathbf{F}$, como pode ser visto em (B-7). Notar também que as restrições em (B-10) são equivalentes ao processo de decodificação Raptor conforme introduzido em (B-3). Os símbolos resultantes $\mathbf{D}$ podem agora ser utilizados como símbolos de entrada para o código Raptor não-sistemático. As restrições impostas pela Equação (B-10) estão representados na Figura B.3, onde (B-8) foi tomado em conta.

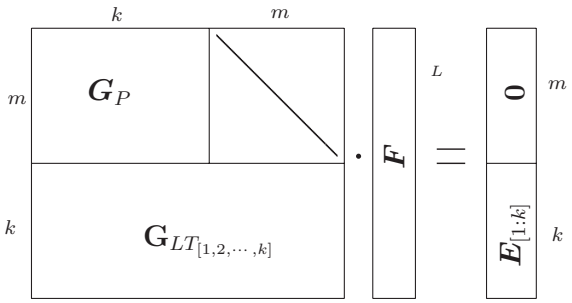

Figura B.3: Restrições na codificação do código Raptor sistemático.

B.4
Implementação dos códigos Raptor não-sistemáticos

Nesta seção vamos descrever como foi realizada a implementação do código Raptor não-sistemático, qual é mostrada mediante um diagrama em blocos na Figura B.4.

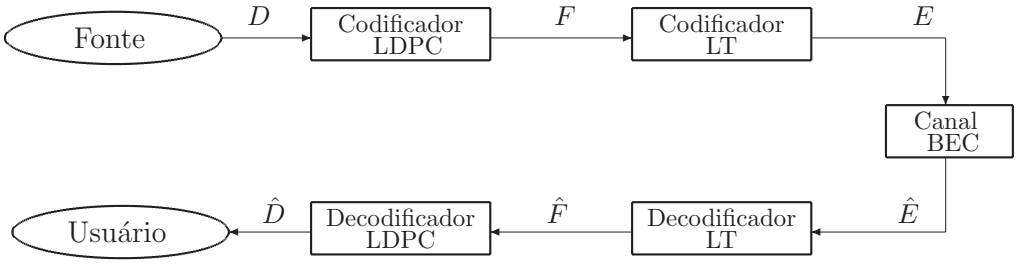

Figura B.4: Diagrama em blocos do código Raptor.

Os códigos Raptor pre-codificam os símbolos fonte utilizando um código de bloco de comprimento fixo, e em seguida codificam esses novos símbolos com um código LT. Este processo de codificação do código Raptor é mostrado graficamente na Figura B.5.

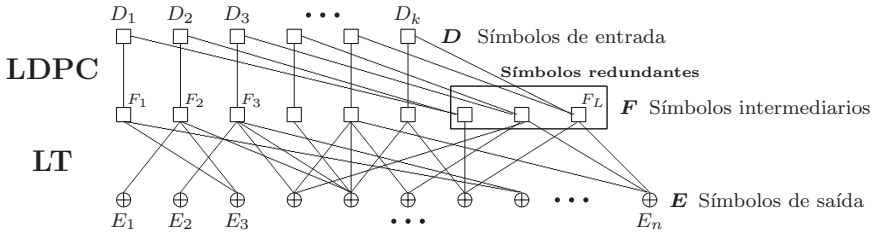

Figura B.5: Processo de codificação do código Raptor.

A principal vantagem é que, para uma decodificação correta, já não é necessário que a decodificação LT tenha sucesso para todos os símbolos pré-codificados. Assim, é possível utilizar uma distribuição de graus mais simples que não recupere todos os símbolos pré-codificados, mas torna-se mais rápido o processo de decodificação. A principal desvantagem dos códigos Raptor é que o *overhead total* é delimitado inferiormente pelo *overhead* do pré-código [3].

O algoritmo de decodificação é composto de duas etapas. Primeiro o decodificador LT interno entrega um vetor com os símbolos pré-codificados recuperados. Logo, este vetor é processado pelo decodificador LDPC externo, baseado no algoritmo *belief propagation* para tentar recuperar os k símbolos de

entrada originais. Shokrollahi propõe em [21] uma nova distribuição de graus que somente depende do *overhead* e tem um grau máximo muito menor que k. Neste trabalho, esta distribuição será chamada *"distribuição de Shokrollahi"* e o novo código LT como *"código LT enfraquecido"* (wLT).

O *overhead total* do código Raptor, ε_{Raptor}, depende dos *overheads* do pré-codificador LDPC e do codificador wLT:

$$(1 + \varepsilon_{Raptor}) = (1 + \varepsilon_{LDPC}) \cdot (1 + \varepsilon_{wLT}). \tag{B-11}$$

A partir da equação anterior, é claro que, dado um *overhead total* como objetivo, existem dois parâmetros para projetar. Como conseqüência, é importante selecionar o melhor par $(\varepsilon_{LDPC}, \varepsilon_{wLT})$ para conseguir uma decodificação eficiente. Shokrollahi [21] sugere ajustar $\varepsilon_{wLT} = 0.5\ \varepsilon_{Raptor}$. Então, a Equação (B-11) é reduzida a uma expressão que relaciona o *overhead total* (ε_{Raptor}) e o *overhead* do pré-código

$$\varepsilon_{Raptor} = \frac{2\ \varepsilon_{LDPC}}{1 - \varepsilon_{LDPC}} \tag{B-12}$$

B.4.1 Construção do pré-código LDPC

Aqui primeiro descrevemos de maneira breve os códigos LDPC e, logo explicamos como é realizado o processo de pré-codificação do código Raptor. Vale mencionar que vamos continuar usando a mesma simbologia que foi utilizada no início deste capítulo.

Os códigos LDPC (do inglês, *low-density parity-check codes*), são códigos de bloco lineares com matriz de paridade $\mathbf{H}$ de dimensão $(L - k) \times L$, onde L é o número de símbolos pré-codificados. A característica que define os códigos LDPC é o fato de que sua matriz de paridade é esparsa, ou seja, o número de valores não nulos é muito menor que o número total de valores na matriz $\mathbf{H}$.

Quanto à regularidade, existem dois tipos de códigos LDPC: regulares e irregulares. Em códigos regulares, todas as linhas e colunas de $\mathbf{H}$ têm o mesmo número de 1s, embora este número possa ser diferente para ambas. Se o número de 1s muda entre colunas e/ou linhas de $\mathbf{H}$, o código é dito irregular.

As características dos códigos LDPC, são descritas em termos de seus grafos de Tanner [23]. Um grafo de Tanner é um grafo bipartido que contém dois tipos de nó: nós de paridade e nós de variável. Qualquer código de bloco

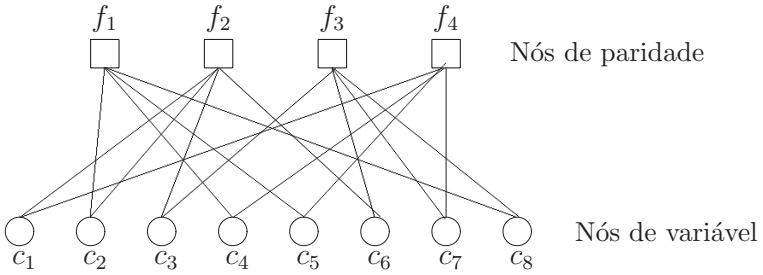

Figura B.6: Grafo de Tanner correspondente à matriz de paridade **H** de (B-13).

linear binário tem um grafo de Tanner correspondente, o qual é construído a partir de sua matriz de paridade **H**. Cada bit na palavra-código corresponde a um nó de variável, e cada equação de paridade corresponde a um nó de paridade. Um nó de paridade f_i é conectado a um nó de variável c_j no grafo de Tanner se, e somente se, o elemento h_{ij} da matriz de paridade **H** é 1. A Figura B.6 mostra o grafo de Tanner associado a um código cuja matriz de paridade é dada por

$$\mathbf{H} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}. \tag{B-13}$$

Em [17], é mostrado que o desempenho de códigos LDPC é determinado pela chamada *"distribuição de densidades"*, que são polinômios que provém uma descrição da estrutura do grafo de Tanner. De fato, define-se o grau de um nó como o número de ramos conectados a ele. Assim, o grau de um nó de variável é o número de equações de paridade do qual o bit correspondente faz parte, enquanto que o grau de um nó de paridade é o número de bits envolvidos na correspondente equação de paridade. Seja λ_i a fração de ramos que estão conectados aos nós de variável de grau i, e seja, ρ_i a fração de ramos que estão conectados aos nós de paridade de grau i. Então,

$$\lambda(x) = \sum_i \lambda_i x^{i-1} \tag{B-14}$$

é a distribuição de graus dos nós de variável, e

$$\rho(x) = \sum_i \rho_i x^{i-1} \tag{B-15}$$

é a distribuição de graus dos nós de paridade.

Para a construção do código Raptor não-sistemático, usamos um código LDPC regular como pré-código $\mathcal{C}_L$, onde o peso das colunas (w_c) e linhas (w_r) da matriz de paridade **H** é constante. A taxa R do pré-código $\mathcal{C}_L$ é dada por

$$R = \frac{\left(1 + \frac{\varepsilon_{Raptor}}{2}\right)}{(1 + \varepsilon_{Raptor})} = 1 - \frac{w_c}{w_r}. \tag{B-16}$$

A matriz de paridade $\mathbf{H}$ do pré-código tem a seguinte forma

$$\mathbf{H} = (\mathbf{H}_1 \mid \mathbf{H}_2) \tag{B-17}$$

onde as submatrices $\mathbf{H}_1$ e $\mathbf{H}_2$ são esparsas. A submatriz $\mathbf{H}_1$ é uma matriz de dimensão $(L-k) \times k$ e a submatriz $\mathbf{H}_2$ é uma matriz quadrada de dimensão $(L-k)\times(L-k)$ não singular, ou seja, que tem matriz inversa $\mathbf{H}_2^{-1}$. É necessário utilizar o método de eliminação Gaussiana, para converter a matriz de paridade original $\mathbf{H}$ em uma matriz de paridade sistemática

$$\mathbf{H}_{SYS} = \left(\mathbf{H}_2^{-1}\mathbf{H}_1 \mid \mathbf{I}_{L-k}\right) = (\mathbf{P} \mid \mathbf{I}_{L-k}) \tag{B-18}$$

onde $\mathbf{I}_{L-k}$ é a matriz identidade de dimensão $(L-k)\times(L-k)$. Então, a matriz geradora do pré-código $\mathbf{G}_{LDPC}$ é uma matriz de dimensão $(k \times L)$ que tem a seguinte forma

$$\mathbf{G}_{LDPC} = \left(\mathbf{I}_k \mid \mathbf{P}^T\right). \tag{B-19}$$

Logo, os $L = k + s$ símbolos pré-codificados denotados pelo vetor $\mathbf{F}$ são gerados da seguinte maneira

$$\mathbf{F} = \mathbf{G}_{LDPC}^T \cdot \mathbf{D}, \tag{B-20}$$

onde os k primeiros símbolos pré-codificados correspondem aos k símbolos de entrada denotados pelo vetor $\mathbf{D}$.

B.4.2 Construção do código LT interno

A construção do código wLT é realizada de maneira similar à construção do código LT convencional que foi estudado no capítulo 2. A única diferença é a distribuição de graus $\Omega_D(x)$ dos símbolos de saída, a qual é muito similar à distribuição Sóliton Ideal, mas tem algumas modificações como ter um grau máximo igual a $D+1$ que é muito menor que o comprimento dos símbolos de entrada do codificador wLT, e dar um peso apropriado para os símbolos de saída de grau 1.

Seja, ε_{Raptor}, o *overhead* do código Raptor, um número real maior que zero. Então definimos a *distribuição de Shokrollahi* como

$$\Omega_D(x) = \frac{1}{\mu+1}\left(\mu x + \frac{x^2}{1\cdot 2} + \frac{x^3}{2\cdot 3} + \cdots + \frac{x^D}{(D-1)\cdot D} + \frac{x^{D+1}}{D}\right), \tag{B-21}$$

onde $D = \lceil 4(1 + \varepsilon_{Raptor})/\varepsilon_{Raptor} \rceil$ e $\mu = (\varepsilon_{Rapor}/2) + (\varepsilon_{Raptor}/2)^2$.

Denotamos à matriz geradora do código wLT, como $\mathbf{G}_{LT}$ de dimensão $(n \times L)$. Logo, os L símbolos intermediários $\mathbf{F}$ são os símbolos de entrada para o codificador wLT para obter os n símbolos codificados $\mathbf{E}$ como

$$\mathbf{E} = \mathbf{G}_{LT} \cdot \mathbf{F}. \tag{B-22}$$

B.5
Resultados obtidos das simulações

Nesta seção são apresentados os resultados das simulações realizadas para comparar o desempenho entre códigos Raptor e códigos LT, calculando a probabilidade de falha na decodificação e a taxa de apagamento de símbolo.

As primeiras simulações foram feitas para determinar a probabilidade de falha na decodificação em função do overhead $(1 + \epsilon)$ para um canal ideal $(P_a = 0)$. Os parâmetros utilizados para cada código são:

- **Código Raptor**: Pré-código LDPC regular com $k = 1000$ símbolos de entrada, taxa $R = (1 + 0.5\ \epsilon)/(1 + \epsilon)$ e $w_c = 3$, e um código wLT com overhead $\varepsilon_{wLT} = \frac{\epsilon}{2}$ e distribuição de graus de Shokrollahi. (Simulações con $w_c = 5$ e $w_c = 7$ conduziram a resultados próximos aos resultados obtidos com $w_c = 3$.)
- **Código LT**: $k = 1000$ símbolos de entrada e distribuição de graus **S**óliton **R**obusta com parâmetros $c = 0.03$, $\delta = 0.1$.

O resultado destas simulações é mostrado na Figura B.7, onde observa-se um melhor desempenho do código Raptor em comparação à código LT com distribuição Sóliton Robusta. Para obter tais resultados foram realizadas 1000 simulações para cada valor de overhead.

Logo, foram realizadas outras simulações, onde foi calculado a taxa de apagamento de símbolo, ou seja, a probabilidade de um símbolo de entrada não ser recuperado após a decodificação de $k(1 + \epsilon)$ símbolos de saída. As simulações foram feitas para um canal BEC com $P_a = 0.04$ e usando os mesmos parâmetros das simulações anteriores. A figura B.8 mostra a taxa de apagamento de símbolo em função do overhead para códigos LT com distribuição SR e códigos Raptor.

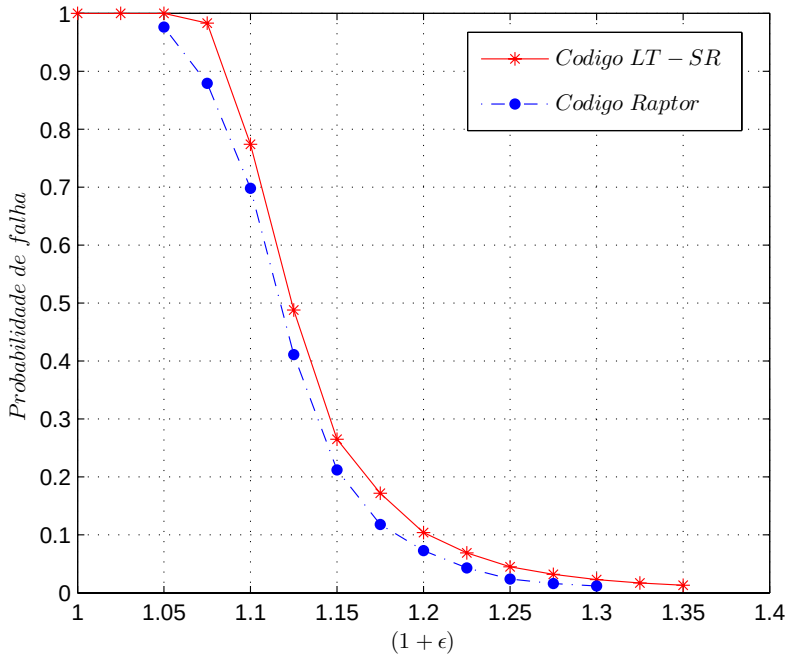

Figura B.7: Probabilidade de falha em função do overhead. Canal ideal.

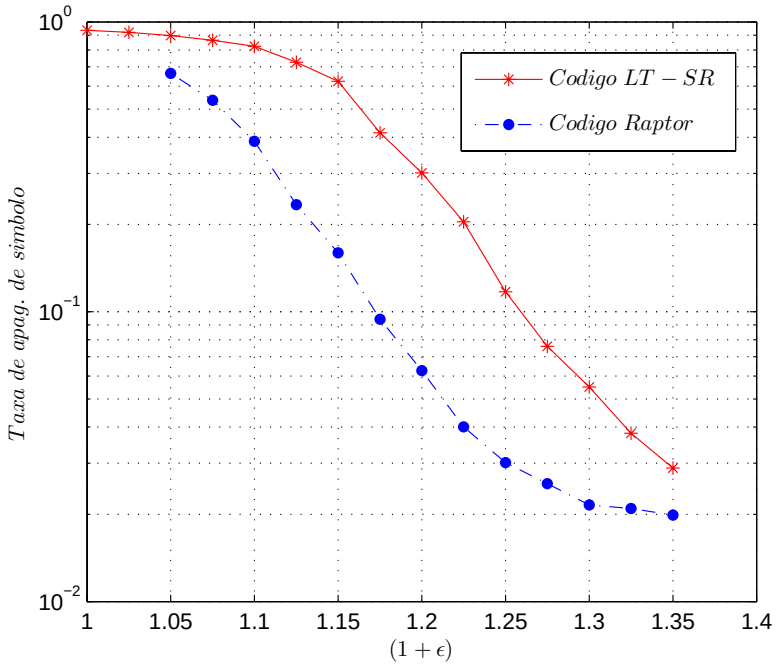

Figura B.8: Taxa de apagamento de símbolo em função do overhead. Canal BEC com $P_a = 0,04$

Finalmente, observou-se através das simulações mostradas na Figura B.9 que os códigos Raptor apresentaram um desempenho inferior ao desempenho dos códigos LT com distribuição SRM.

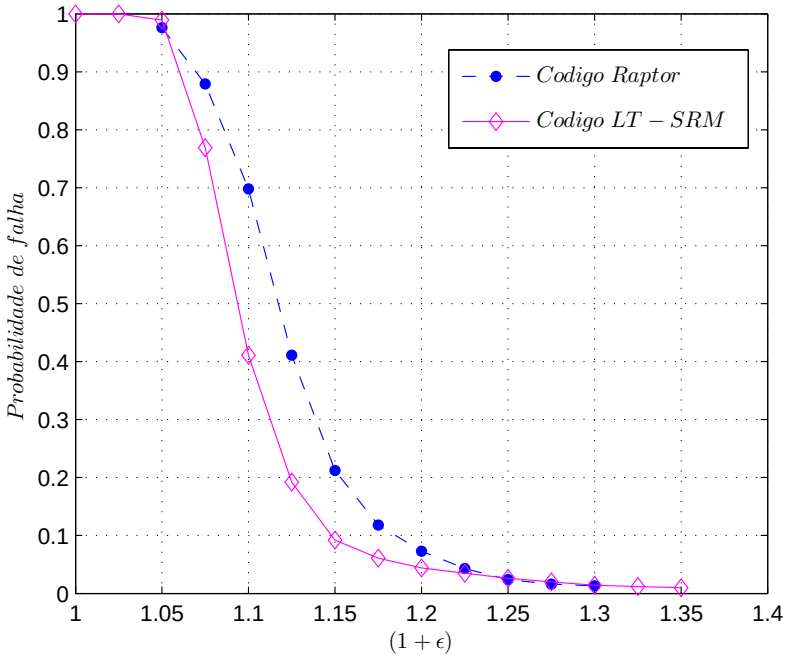

Figura B.9: Comparação de desempenho entre códigos LT com distribuição SRM e códigos Raptor. Canal Ideal.

Este fato é conhecido na literatura, que o desempenho de códigos Raptor construídos com pré-codificador LDPC irregular tenha desempenho melhor que os códigos LT/SRM.