

# Proposta de um decodificador para a codificação *Two Dimensional and Two Diagonal Parity Check*

Ana Júlia Fontoura, Tauame Aguiar Pacce

Departamento de Informática – Universidade de Santa Cruz do Sul (UNISC)  
CEP 96810-900 – Santa Cruz do Sul – RS – Brasil  
{ana.fontoura,tauamepacce}@hotmail.com

**Abstract.** *This paper describes the 2D2D (Two Dimensional and Two Diagonal Parity Check) coder, performing also an analysis of the same. In addition to this analysis, it is proposed a decoder with the intent to prove the coded data can be recovered, even in the presence of noise.*

**Resumo.** *Este artigo descreve o codificador 2D2D (Two Dimensional and Two Diagonal Parity Check), realizando também uma análise do mesmo. Além desta análise, propõe-se um decodificador com o objetivo de provar que os dados assim codificados podem ser recuperados, mesmo na presença de ruído.*

## 1. Introdução

Existem duas formas básicas de se tratar erros na transmissão de pacotes de dados, segundo [Tanenbaum 2003]. Uma delas é adicionar *bits* redundantes à informação que está sendo enviada. A quantidade de *bits* adicionados deve ser suficiente para que se possam detectar possíveis erros que tenham ocorrido durante a transmissão, e solicitar um reenvio do pacote. Outra técnica, chamada de *FEC (Forward Error Correction)*, se baseia na adição de um número maior de *bits* de redundância que a anterior, mas não somente com o objetivo de detectar o erro, e sim de também corrigi-lo.

A escolha entre as técnicas depende da confiabilidade no canal onde a informação será transmitida, e também do *delay* de transmissão devido às grandes distâncias [Stallings 2004]. Por exemplo, para enlaces de satélites, como o tempo necessário para reenviar um pacote é muito grande, opta-se pela correção do erro, ao invés da detecção e reenvio.

Assim, este artigo tem como objetivo criar um decodificador para a codificação *2D2D (Two Dimensional and Two Diagonal parity check)* proposta por [Aflakian 2011], afim de provar que esta codificação pode ser decodificada em tempo polinomial. Este trabalho está organizado da seguinte forma: na Seção 2 é descrito o funcionamento do codificador *2D2D*; na Seção 3 é feita uma análise da codificação *2D2D*; na Seção 4 é proposto um algoritmo decodificador para o código; e finalmente, na Seção 5 são apresentadas as conclusões.

## 2. Codificador

O *FEC 2D2D* é um codificador do tipo de blocos, que utiliza *bits* de paridade em uma matriz bidimensional para realizar a detecção e correção de erros [Aflakian 2011].

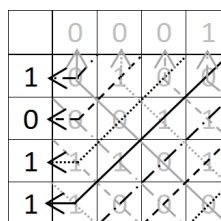
Embora a codificação proposta por [Aflakian 2011] possua características de *overhead* e *code rate* interessantes, em nenhum momento o autor descreveu um decodificador que realizasse a decodificação em tempo polinomial.

A codificação é realizada de maneira que em cada bloco, os dados são dispostos em uma matriz de L linhas por L colunas. Então uma linha de paridade, com largura L, é adicionada no topo da matriz. O valor que os *bits* dessa linha irão assumir, é decidido realizando uma paridade do tipo par nas diagonais da matriz (no sentido topo-esquerda até base-direita), conforme a Figura 1. As diagonais que vêm depois da diagonal principal são divididas em duas partes: a primeira é a diagonal diretamente abaixo do *bit* de paridade, e a outra é a diagonal ainda não utilizada mais distante abaixo da diagonal principal.



**Figura 1. Linha de paridade do topo**

É adicionado também uma coluna de paridade no lado esquerdo da matriz, que segue a mesma lógica da primeira linha de paridade, porém com as diagonais no sentido base-esquerda até topo-direita, conforme a Figura 2. Depois, são adicionadas também uma coluna de paridade a direita da matriz que é formada pela paridade das linhas da matriz; e uma linha de paridade na base da matriz que é formada pela paridade das colunas da matriz.



**Figura 2. Coluna de paridade da esquerda**

Enfim, [Aflakian 2011] afirma que são adicionados *bits* de paridade nas quinas da matriz, conforme a Figura 3. Estes *bits* são construídos da seguinte forma: o *bit* da base-esquerda da matriz é formado pela paridade da coluna esquerda de paridade (das diagonais sentido base-esquerda até topo-direita). O *bit* do topo-direita é formado pela linha de paridade do topo (das diagonais sentido topo-esquerda até base-direita). O *bit* da base-direita é formado pela linha de paridade da base (das colunas da matriz) mais o *bit* de quina base-esquerda calculado anteriormente. O *bit* do topo-esquerda é formado tomando a paridade de toda a coluna de paridades da direita (das linhas da matriz), incluindo também as duas quinas calculadas do lado direito da matriz. Este último *bit* de quina (topo-esquerda) não possui nenhum outro *bit* que checa sua validade, e por tanto é um ponto de falha da codificação. Para mitigar este problema, [Aflakian 2011] sugeriu

como solução repetir este *bit* três vezes para garantir que ele chegue corretamente ao receptor.

0	0	0	0	1	1
1	0	1	0	0	1
0	0	0	1	1	0
1	1	1	0	1	1
1	1	0	0	0	1
1	0	0	1	0	0

Figura 3. Matriz 2D2D com os *bits* de paridade

### 3. Análise do codificador 2D2D

Ao observar a matriz resultante da codificação 2D2D, pode-se afirmar que a *code rate*  $CR$  é resultado direto da largura da matriz de dados  $L$ , conforme a fórmula

$$CR = \frac{L^2}{L^2 + 4L + 6}.$$

Segundo [Aflakian 2011], este código possui o mesmo *overhead* da

codificação *four dimensional parity check* proposta por [Naveen 2004], que é capaz de corrigir até 2 erros. Apesar de possuírem o mesmo *overhead*, [Aflakian 2011] afirma que a 2D2D tem a capacidade de corrigir até 4 erros de *bits*. Porém, o que foi observado na prática é que ela corrige quaisquer erros de 3 *bits* e a maioria dos erros de 4 *bits*. Utilizando o decodificador descrito na próxima seção, foi possível em alguns poucos casos corrigir até 5 erros em um bloco, mas na maioria das vezes o que acontece é que a correção transforma o bloco errado em um outro bloco válido, diferente do transmitido.

Embora matrizes bidimensionais tenham a capacidade de corrigir apenas um erro de bit segundo [Kurose 2003], esta modificação adicionando uma linha e uma coluna de paridade para as diagonais resulta em um aumento significativo na capacidade de correção da mesma. Considerando uma capacidade de correção de erros de 4 *bits* por bloco, construiu-se o gráfico da Figura 4 que demonstra a relação entre a largura da matriz de dados, a *code rate* e a capacidade de correção de erros.

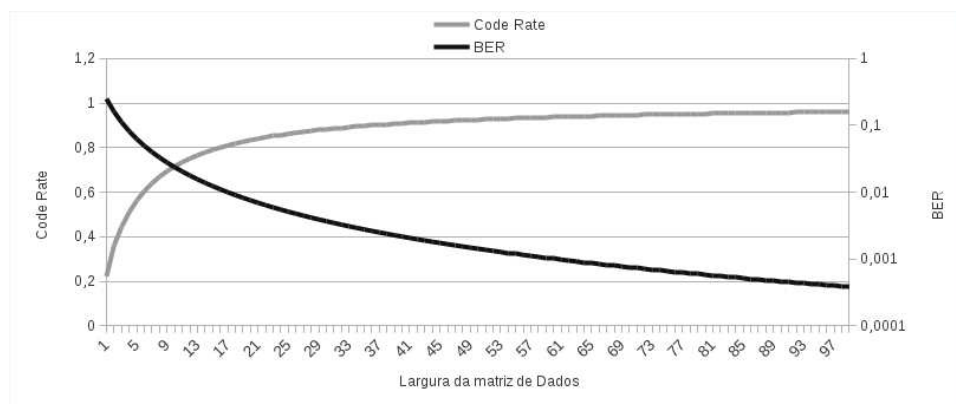
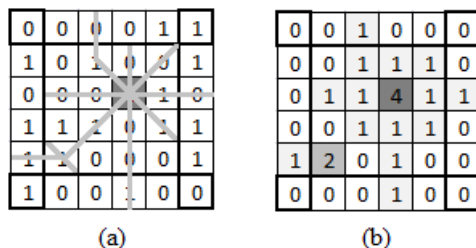


Figura 4. Gráfico *Code Rate* x BER x Largura Matriz de Dados

#### 4. Decodificador

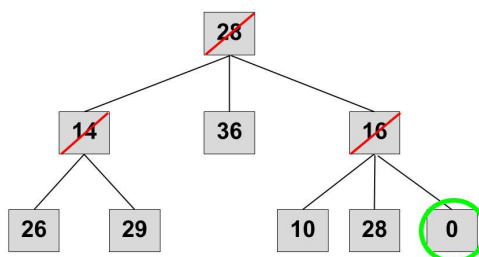
Devido ao fato de o autor da codificação 2D2D não ter previsto nenhum decodificador para o seu trabalho, decidiu-se por construir um decodificador baseado em espaço de buscas em árvores, a fim de provar que é possível realizar uma decodificação em tempo polinomial do código proposto. Na árvore de buscas implementada, cada nodo representa uma configuração da matriz do bloco, e os nodos filhos são as matrizes que são obtidas ao permutar o estado de um único *bit* da matriz. Cada nodo da árvore possui uma pontuação que é utilizada para escolher qual o próximo nodo que será expandido. Esta pontuação é baseada no número de marcadores de erro no nodo e quanto mais próximo de 0, mais próximo da solução o nodo se encontra.

Cada *bit* de paridade que não estiver de acordo com as células da qual ele é a paridade, resulta em um marcador de erro em cada célula do conjunto. Uma célula pode possuir mais de um marcador de erro se ela tiver mais *bits* de paridade que não estão de acordo, como pode-se observar na Figura 5, onde uma das células possui 4 marcadores de erro, e outra possui 2, enquanto diversas possuem apenas 1 ou nenhum.



**Figura 5. Bit invertido em uma matriz 2D2D (a) ao lado dos respectivos marcadores de erro gerados pelo mesmo (b).**

O algoritmo busca nos nodos folha da árvore pelo nodo com a menor pontuação (menor número de marcadores de erro) e expande este nodo. O processo de expandir o nodo se baseia em selecionar as células da matriz com maior número marcadores de erro, pois estas têm mais chances de serem os *bits* errados. As células selecionadas têm então o seu bit invertido, gerando uma nova matriz que será um nodo filho do nodo atual. O processo então de selecionar um novo nodo folha para expandir é repetido até que se tenha atingido um nodo com pontuação 0. Um exemplo de busca pode ser visto na Figura 6.



**Figura 6. Árvore de busca com número de marcadores de erro em cada nodo.**

Se através dos passos descritos anteriormente, o algoritmo expandir todos os nodos da árvore em até 4 níveis e mesmo assim não encontrar um nodo solução (com

pontuação 0), o algoritmo para a execução e informa que a solução não foi encontrada. Outro critério de parada que foi adotado arbitrariamente para evitar consumo excessivo de memória é o caso onde mais de 1000 nodos são expandidos e mesmo assim a solução não é encontrada. Ambos os critérios de parada podem ser alterados, para que o algoritmo se adeqüe as limitações do *hardware* disponível.

## 5. Conclusão

Depois de implementado o codificador e o decodificador *2D2D*, pode-se concluir que apesar de o bloco ser formado por uma matriz bidimensional, e possuir o mesmo *overhead* de uma codificação *four dimensional parity check*, a *2D2D* é capaz de corrigir até duas vezes mais erros do que a outra. Verificou-se também que a codificação é melhor aproveitada com larguras de matriz de dados de 10 a 20 *bits*, pois nestes casos obtiveram-se os melhores resultados de desempenho.

Finalmente, o decodificador proposto cumpriu o objetivo de provar que a codificação *2D2D* pode ser decodificada, porém a complexidade temporal e a viabilidade desta solução ainda precisam ser estudadas mais a fundo.

## Referências

- Tanenbaum, A. S. (2003) “Redes de Computadores”, 4ª Ed. Elsevier Editora Ltda. p. 205-206.
- Stallings, W. (2004) “Data and Computer Communications” 7ª Ed., Estados Unidos, Pearson Education, Inc. p. 185-186.
- Aflakian, D., Siddiqui, T. *et al.* (2011) “Error Correction and Error Detection over Two-Dimensional and Two-Diagonal Model and Five-Dimensional Model”, International Journal of Advanced Computer Science and Applications, Vol. 2, No. 7.
- Naveen, B. A., Thirunavukkarasu, U. e Latifi, S. (2004) “Three and Four-dimensional Parity-check Codes”, Proceedings of the International Conference on Information Technology: Coding and Computing.
- Kurose, J. F. e Ross, K. W. (2003) “Computer Networking: A Top-Down Approach Featuring the Internet” 2ª Ed., Estados Unidos, Pearson Education Inc. p. 427-428.