

ROBERTO WILHELM KRAUSS MARTINEZ

**CORREÇÃO DE ERROS NA CAMADA DE APLICAÇÃO  
PARA RECEPÇÃO MÓVEL DO SISTEMA BRASILEIRO DE  
TELEVISÃO DIGITAL**

CURITIBA

2012

ROBERTO WILHELM KRAUSS MARTINEZ

**CORREÇÃO DE ERROS NA CAMADA DE APLICAÇÃO  
PARA RECEPÇÃO MÓVEL DO SISTEMA BRASILEIRO DE  
TELEVISÃO DIGITAL**

Dissertação apresentada como requisito parcial  
à obtenção do grau de Mestre. Programa de  
Pós-Graduação em Engenharia Elétrica, Setor  
de Tecnologia, Universidade Federal do Paraná.

Orientador: Prof. Dr. Evelio Martín Gar-  
cía Fernández

CURITIBA

2012

## AGRADECIMENTOS

Primeiramente a Deus.

Ao meu orientador, Prof. Dr. Evelio Martín Garcia Fernández pelas suas importantes contribuições, dedicação, paciência, ajuda e oportunidade.

A minha esposa Lívia Hartmann pelo amor, carinho, apoio e paciência, especialmente nas intermináveis horas de correções. A conquista é nossa!

A meus pais e irmãos, Mary Stella Martinez e Gunther Krauss Emhart, Germán Gabriel Krauss Martinez e Christofer Jose Maria Krauss Martinez, pelo amor, carinho, suporte e pelo exemplo de vida, dedicação e trabalho.

Ao meus colegas da UFPR: Rodrigo Seger, Christofer Shwartz, Fernando Zanella, Samuel Baraldi Mafra, Daniel Cavalcanti, Jorge Angeloni e Marcelo Luiz de Souza pelos bons momentos passados e pelo apoio oferecido.

Agradeço ao corpo docente do PPGEE da UFPR, pelo ensino recebido.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes).

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ).

## SUMÁRIO

<b>LISTA DE FIGURAS</b>	<b>vi</b>
<b>LISTA DE TABELAS</b>	<b>vii</b>
<b>LISTA DE SIGLAS</b>	<b>ix</b>
<b>RESUMO</b>	<b>x</b>
<b>ABSTRACT</b>	<b>xi</b>
<b>1 INTRODUÇÃO</b>	<b>1</b>
1.1 Motivação e Justificativa . . . . .	1
1.2 Objetivos . . . . .	3
1.3 Metodologia de Construção . . . . .	3
1.4 Estrutura da Dissertação . . . . .	4
<b>2 O SISTEMA BRASILEIRO DE TELEVISÃO DIGITAL - SBTVD</b>	<b>6</b>
2.1 Introdução . . . . .	6
2.2 <i>Transport Stream</i> TS . . . . .	8
2.3 Entrelaçamento no Tempo e na Frequência do SBTVD . . . . .	10
2.4 Considerações Finais . . . . .	12
<b>3 CÓDIGOS FONTANAIS</b>	<b>13</b>
3.1 Introdução . . . . .	13
3.2 Códigos LT . . . . .	13
3.2.1 Codificação LT . . . . .	14
3.2.2 Codificador LT como um Código em Grafo . . . . .	14
3.2.3 Decodificação LT . . . . .	16
3.2.4 Distribuição de Graus . . . . .	17

3.3	Códigos de Raptor . . . . .	21
3.3.1	Códigos de Raptor Não-Sistemáticos . . . . .	21
3.3.2	Codificação Raptor Sistemática . . . . .	23
3.3.3	Matriz de Pré-Codificação . . . . .	25
3.3.4	<i>Overhead</i> do Código Raptor . . . . .	26
3.3.5	Código Interno LT . . . . .	27
3.4	Considerações Finais . . . . .	27
<b>4</b>	<b>METODOLOGIA</b>	<b>29</b>
4.1	Introdução . . . . .	29
4.2	Códigos Fontanais . . . . .	31
4.2.1	Pré-código LDPC para Código Raptor . . . . .	31
4.2.2	Distribuição de Graus do LT para o Código Raptor . . . . .	31
4.2.3	Metodologia para Decodificação dos Códigos Fontanais . . . . .	33
4.2.4	Distribuição de Graus para o Código LT . . . . .	33
4.3	Tempo de Proteção . . . . .	34
4.4	O Simulador . . . . .	35
4.5	Modelos de Canais Implementados no Simulador . . . . .	37
4.5.1	Decodificador Externo de Reed Solomon . . . . .	39
4.6	Modelo de Cadeia de Markov de Dois Estados . . . . .	40
4.7	Considerações Finais . . . . .	47
<b>5</b>	<b>RESULTADOS</b>	<b>48</b>
5.1	Introdução . . . . .	48
5.2	Comparação entre Código Raptor e Código LT . . . . .	48
5.3	Desempenho do Código Raptor no SBTVD Móvel . . . . .	49
5.4	Taxa de Decodificação com Sucesso . . . . .	52
<b>6</b>	<b>CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS</b>	<b>55</b>
	<b>BIBLIOGRAFIA</b>	<b>60</b>



## LISTA DE FIGURAS

1.1	Comparação de desempenho do SBTVD para o canal Brasil A estático e móvel . . . . .	1
1.2	Arquitetura do sistema proposto . . . . .	4
2.1	Configuração das camadas para os 13 segmentos . . . . .	7
2.2	Diagrama representativo do encapsulamento do TS . . . . .	9
3.1	Grafo de codificação . . . . .	15
3.2	Processo de decodificação . . . . .	17
3.3	Distribuição Sóliton Robusta para $k = 10000$ , $c = 0,2$ e $\delta = 0,1$ . . . . .	19
3.4	Histograma do número de símbolos de <i>overhead</i> necessários para uma decodificação com sucesso para $k = 1000$ e distintos valores de $c$ e $\delta$ . . . . .	20
3.5	Estágio de codificação do Código Raptor . . . . .	21
3.6	Diagrama em bloco da Codificação Raptor Sistemática Conceptual . . . . .	24
4.1	Arquitetura do sistema proposto . . . . .	30
4.2	Comparação das distribuições de graus do LT para o Raptor aplicado ao SBTVD móvel, utilizando-se a Equação 3.19, e a tabela apresentada por de Shokrollahi (2006) . . . . .	32
4.3	Taxa de Erro de <i>Bit</i> vs $E_bN_0$ para Taxa do Código Convolutacional 1/2, Comprimento do <i>Interleaving</i> 4, Canal Brasil A $f_d = 34 H_z$ . . . . .	40
4.4	Modelo Gilbert-Elliot, ou modelo de Cadeia de Markov de Dois Estados . . . . .	41
4.5	Modelamento dos resultados obtidos no Simulador de Schwartz com a ferramenta de Gilbert-Elliot . . . . .	42
4.6	Intervalo de Confiança para $E_bN_0 = 10$ . . . . .	43
4.7	Intervalo de Confiança para $E_bN_0 = 13$ . . . . .	44
4.8	Intervalo de Confiança para $E_bN_0 = 15$ . . . . .	44
4.9	Intervalo de Confiança para $E_bN_0 = 17$ . . . . .	45

4.10	Intervalo de Confiança para $EbN0 = 20$ . . . . .	45
4.11	Intervalo de Confiança para $EbN0 = 22$ . . . . .	46
5.1	Desempenho dos Códigos Fontanais Raptor e LT, para um <i>overhead</i> de 20% do tamanho dos símbolos fonte, com $c = 0,03$ e $\delta = 0,1$ na distribuição Sóliton Robusta . . . . .	49
5.2	Desempenho do Código Raptor para um <i>overhead</i> de 20% do tamanho dos símbolos fonte. A curva corresponde à Taxa do Código Convolutacional 1/2, Comprimento do <i>Interleaving</i> 4 e Canal Brasil A $f_d = 10$ Hz . . . . .	50
5.3	Desempenho do Código Raptor para um <i>overhead</i> de 50% do tamanho dos símbolos fonte. A curva corresponde à Taxa do Código Convolutacional 1/2, Comprimento do <i>Interleaving</i> 4 e Canal Brasil A $f_d = 34$ Hz . . . . .	51
5.4	Desempenho do Código Raptor para um <i>overhead</i> de 50% do tamanho dos símbolos fonte. A curva corresponde à Taxa do Código Convolutacional 2/3, Comprimento do <i>Interleaving</i> 4 e Canal Brasil A $f_d = 34$ Hz . . . . .	52
6.1	Medições canal 22 Mercosul . . . . .	61
6.2	Medições canal 22 Mercosul . . . . .	62
6.3	Medições canal 34 (RIC) Rede Record . . . . .	63
6.4	Medições canal 34 (RIC) Rede Record . . . . .	64
6.5	Medições canal 25 Rede Vida . . . . .	65
6.6	Medições canal 25 Rede Vida . . . . .	66
6.7	Medições canal 38 Bandeirantes . . . . .	67
6.8	Medições canal 38 Bandeirantes . . . . .	68
6.9	Medições canal 41 Rede Paranaense de Comunicação TV . . . . .	69
6.10	Medições canal 41 Rede Paranaense de Comunicação TV . . . . .	70



## LISTA DE TABELAS

2.1	Atrasos do entrelaçador temporal. . . . .	11
3.1	Representação do grafo biparticionado . . . . .	15
4.1	Graus de distribuição para o código interno LT para diferentes valores de $k$	32
4.2	Especificação de atraso a ser considerada nas simulações . . . . .	34
4.3	Parâmetros de transmissão das emissoras no Brasil . . . . .	36
4.4	Especificações de atraso e atenuação para os diferentes canais Brasil . . . . .	38
4.5	Erro entre o ponto máximo e o ponto mínimo do intervalo de confiança e o valor entregue pelo Simulador . . . . .	46
5.1	Taxa máxima para Canal Brasil A, Frequência Doppler de 10 Hz . . . . .	53
5.2	Taxa máxima para Canal Brasil A, Frequência Doppler de 34 Hz, Código Convolutacional 1/2 . . . . .	53
5.3	Taxa máxima para Canal Brasil A, Frequência Doppler de 34 Hz, Código Convolutacional 2/3 . . . . .	54

## LISTA DE SIGLAS

3GPP	3rd Generation Partnership Project
ABNT	Associação Brasileira de Normas Técnicas
AL-FEC	Application Layer Forward Error Correction
AL-FEC	Application Layer Forward Error Correction
BEC	Binary Erasure Channel
BER	<i>Bit</i> Error Rate
DVB-H	Digital Video Broadcasting - Handheld
DVB-T	Digital Video Broadcasting - Terrestrial
DVB	Digital Video Broadcasting
$E_b/N_0$	Taxa de Energia de <i>Bit</i> por Densidade Espectral do Ruído
ES	Elementary Stream
HDTV	High Definition Television
IBGE	Instituto Brasileiro de Geografia e Estatística
ISDB-T	Integrated Services Digital Broadcasting - Terrestrial
LDGM	Low Density Generator Matrix
LDPC	Low Density Parity Codes
LT	Luby Transform
Matlab	Matrix Laboratory
MBMS	Multimedia Broadcast Multicast Services
MPE-FEC	Multi Protocol Encapsulation - Forward Error Correction
NIT	Network Information Table
OFDM	Orthogonal Frequency-Division Multiplexing
One-Seg	One Segment

PAT	Program Association Table
PCR	Program Clock Reference
PEG	Progressive Edge Growth
PES	Packetized Elementary Stream
PID	Program Identification
PMT	Program Map Table
Raptor	Rapid Tornado
RS	Reed-Solomon
SBTVD	Sistema Brasileiro de Televisão Digital
SDT	Service Description Table
SDTV	Standard Television Definition
SI	System Information
TEI	Transport Error Indicator
TEI	Transport Error Indicator
TOT	Time Offset Table
TS	Transport Stream
TSP	Transport Stream Packet

## RESUMO

Neste trabalho, pesquisa-se o uso do da correção de erro na camada de aplicação aplicado ao SBTVD (Sistema Brasileiro de Televisão Digital) para recepção móvel. O SBTVD é um padrão de Televisão Digital adotado pelo Brasil e ultimamente adotado por alguns países da América do Sul. Este padrão foi desenvolvido para recepção fixa e portátil. Porém, a recepção móvel não é tão eficaz devido ao Efeito Doppler que ocorre quando o usuário está se movimentando. Visando não modificar as normas e o *hardware* já estabelecidos, apresenta-se a oportunidade de estudar-se a correção de erros na camada de aplicação AL-FEC (do inglês: *Application Layer Forward Error Correction*). Propõe-se assim, a avaliação da simulação do desempenho da implementação do AL-FEC baseado nos Códigos LT e Raptor aplicados a um simulador do SBTVD que reproduz o sistema de transmissão e de codificação de canal. Simulações apresentadas mostram que os Códigos Raptor apresentam um bom desempenho no SBTVD para recepção de aparelhos móveis.

**Palavras chave:** SBTVD, Códigos Raptor, Códigos LT, SBTVD recepção móvel, AL-FEC

## ABSTRACT

This work proposes the use of an application layer forward error correction implemented to the SBTVD for mobile reception. The SBTVD is a digital television TV standard adopted by Brazil and some other countries in South America. This standard was developed for fixed and mobile reception. However mobile reception is not as effective due to the Doppler Effect that occurs when the user is on movement. In order not to modify the standards and hardware already established, the opportunity to study the application layer forward error correction ALFEC has been presented. Therefore it is proposed the simulation evaluation of ALFEC performance based on Raptor and LT codes applied in a SBTVD Simulator which recreates the transmission and channel coding of SBTVD. Simulations presented show that Raptor Codes perform well, better than LT Codes in SBTVD for mobile reception.

**Key words:** SBTVD, Raptor Codes, LT Codes, SBTVD mobile reception, AL-FEC

## CAPÍTULO 1

### INTRODUÇÃO

#### 1.1 Motivação e Justificativa

Segundo o Fórum do Sistema Brasileiro de TV Digital Terrestre, o sinal de TV digital aberta é recebido gratuitamente por 457 municípios no Brasil. De acordo com dados do censo de 2010, realizado pelo IBGE (Instituto Brasileiro de Geografia e Estatística), isto equivale a 45,4% da população brasileira. Pode-se dizer que como aconteceu em países como os Estados Unidos, a transmissão analógica será substituída pela transmissão digital.

O SBTVD (Sistema Brasileiro de Televisão Digital) é um padrão de Televisão Digital adotado pelo Brasil e ultimamente adotado por alguns países da América do Sul. Este padrão foi desenvolvido para recepção fixa e portátil. Porém, a recepção móvel não é tão eficaz devido ao Efeito Doppler e o sombreamento que ocorrem quando o usuário está se movimentando, figura 1.1.

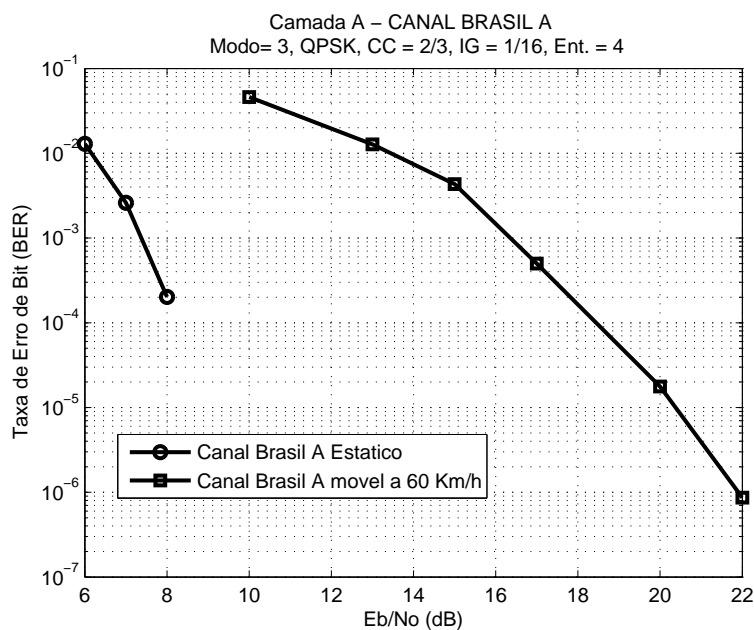


Figura 1.1: Comparação de desempenho do SBTVD para o canal Brasil A estático e móvel

Visando não modificar as normas e o *hardware* já estabelecidos, apresentou-se a opor-

tunidade de estudar-se a correção de erros na camada de aplicação AL-FEC (do inglês: *Application Layer Forward Error Correction*). Assim, as atualizações serão incorporadas como atualização de *software* nos sistemas pré-existentes.

Os códigos mais estudados para utilização na camada de aplicação são: Reed-Solomon [1], Fontanais [2] e LDPC [3], para AL-FEC.

No trabalho apresentado por Iliev et al. (2008)[1], discutiu-se a implementação de Códigos de RS (Reed-Solomon) na camada de aplicação, para padrão europeu de TV digital DVB (DVB do inglês: *Digital Video Broadcasting*). Apesar da vantagem de serem muito eficientes, esses códigos apresentam complexidade de cálculo de codificação e decodificação muito elevada, deixando as execuções das operações muito demoradas para qualquer tipo de processador.

O trabalho de Cunche e Roca (2008)[3] apresenta os Códigos LDPC - LDGM *Staircase* (LDPC - LDGM do inglês: *Low Density Parity Codes - Low Density Generator Matrix*) e LDGM *Triangle*, mostrando que os códigos trabalham com blocos grandes de símbolos. Os autores também provaram a vantagem de velocidade existente no código LDPC *Triangle* quando comparado ao Reed-Solomon. Eles observaram que o LDPC *Staircase* é menos complexo que o LDPC *Triangle*, mas, por outro lado, ele fica mais afastado do ideal que o LDPC *Triangle*.

Montalban Gutierrez (2009) [4], apresenta a comparação dos Códigos LDPC-LDGM com os Códigos Raptor (Raptor do inglês: *Rapid Tornado*), para situações em que são necessários blocos grandes de codificação, e mostra que os Códigos Raptor tem melhor desempenho em quase todos os cenários.

A principal motivação desta pesquisa é o trabalho apresentado por Gonzálvez et al. (2009)[2], onde os autores pesquisaram a utilização do Código de Raptor (da família dos Códigos Fontanais) na camada de aplicação para o Sistema de Televisão Digital Europeu DVB-T (DVB-T do inglês: *Digital Video Broadcasting - Terrestrial*) para provisão de serviços móveis, e comparou-o com o código já implementado no DVB-H (DVB-H do inglês: *Digital Video Broadcasting - Handheld*), o MPE-FEC (MPE-FEC do inglês: *Multi Protocol Encapsulation - Forward Error Correction*).

O MPE-FEC foi desenhado no sistema DVB-T para lidar com o desvanecimento rápido do canal, além de realizar um entrelaçamento virtual da informação transmitida.

Gozálvez et al. (2009)[2] concluíram que aumentando o tempo de proteção, obtêm-se ganhos de 1 até 7 dB comparados ao MPE-FEC, incluindo os ganhos de diversidade espacial.

No sistema DVB-H, os Códigos de Raptor foram previamente padronizados para a proteção no nível de enlace segundo as normas ETSI *Bluebook* A131 (2008) [5] e ETSI *Bluebook* A054r4.1 (2009)[6].

A partir do trabalho apresentado por González et al. (2009)[2] e partindo do ponto de vista que algumas características do Sistema Europeu DVB-T são parecidas ao SBTVD, optou-se pelo estudo dos Códigos Raptor e, conseqüentemente, os Códigos fontanais na camada de aplicação empregado ao SBTVD móvel.

## 1.2 Objetivos

O objetivo do projeto é melhorar o desempenho na recepção móvel do SBTVD através da utilização de um código corretor de erros na camada de aplicação.

## 1.3 Metodologia de Construção

Desafortunadamente, o uso da diversidade espacial, ou uso de múltiplas antenas, é visto como um problema nos aparelhos móveis devido à limitação de tamanho, potência e complexidade. Uma saída para melhorar a qualidade de serviço no sistema móvel é recorrer à diversidade temporal, ou uso de códigos corretores de erro. Com a finalidade de que seja implementada sem a atualização do *hardware*, foi escolhida o uso de códigos corretores de erro na camada de aplicação, com a vantagem que somente será necessária a atualização de *software*.

A figura 1.2 mostra a arquitetura proposta para este trabalho. Como se pode observar o áudio e o vídeo são codificados na camada de aplicação pelo AL-FEC, gerando um pacote de redundância FEC. Estes pacotes: o de áudio, vídeo e FEC são empacotados num fluxo



de transporte TS (TS do inglês: *Transport Stream*) e transmitido pelo SBTVD ou ISDB-T International (ISDB-T International do inglês: *Integrated Services Digital Broadcasting - Terrestrial International*).

Os receptores que não suportam a arquitetura proposta simplesmente descartariam o pacote de redundância. Por outro lado, os receptores que contêm a arquitetura proposta aproveitariam o pacote de redundância FEC, fariam uma decodificação na camada de aplicação e corrigiriam os pacotes errados.

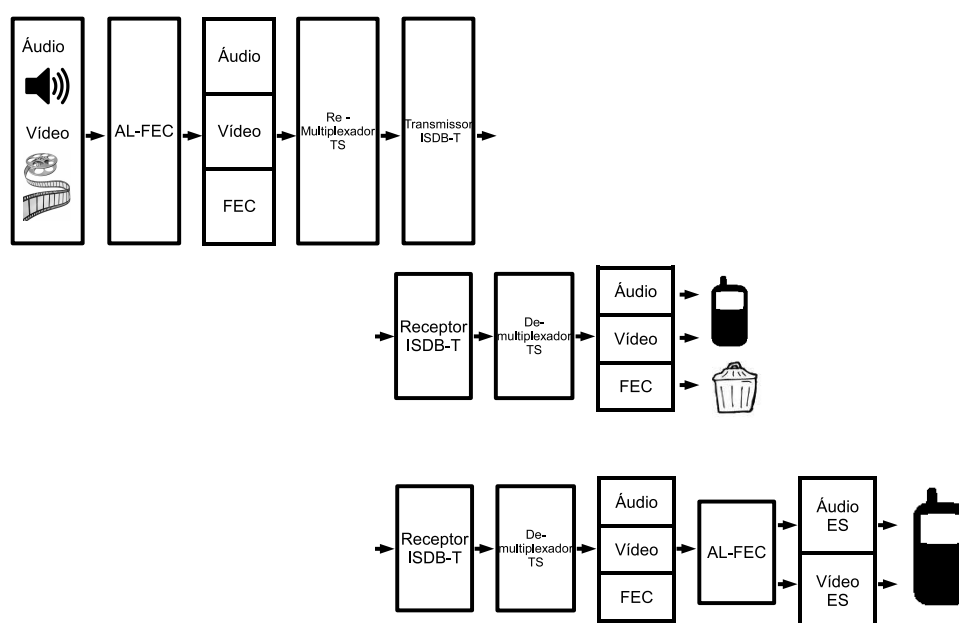


Figura 1.2: Arquitetura do sistema proposto

Para todos os casos neste trabalho, serão realizadas simulações computacionais no *software* Matlab (Matlab do inglês: *Matrix Laboratory*)

## 1.4 Estrutura da Dissertação

A presente dissertação está estruturada da seguinte maneira: o Capítulo 2 trata do SBTVD e apresenta as principais propriedades relevantes para este trabalho. No Capítulo 3, serão apresentadas as principais características e propriedades dos Códigos Fontanais como métodos de construção, codificação e decodificação. No Capítulo 4, serão apresen-

tadas considerações sobre a metodologia envolvida para chegar ao objetivo proposto neste trabalho. No Capítulo 5, serão mostrados os resultados obtidos da comparação dos Códigos Fontanais e o desempenho dos mesmos no SBTVD. Por fim, o Capítulo 6 apresenta as conclusões finais, além de sugestões para trabalhos futuros.

## CAPÍTULO 2

# O SISTEMA BRASILEIRO DE TELEVISÃO DIGITAL - SBTVD

### 2.1 Introdução

Uma vez que o cenário deste trabalho é o SBTVD móvel, neste capítulo serão apresentadas partes relevantes do SBTVD para o sistema móvel, desprezando os detalhes do estudo aprofundado, dada a correção de erro na camada de aplicação.

O espectro do SBTVD deve consistir-se obrigatoriamente em 13 segmentos OFDM sucessivos, dentro de uma banda de 6 MHz, a qual deve permitir a transmissão hierárquica, ou seja, permitir a recepção fixa HDTV (HDTV do inglês: *High Definition Television*), SDTV (SDTV do inglês: *Standard Television definition*) e móvel (portátil - One-Seg (One-Seg do inglês: *One Segment*)).

Os 13 segmentos podem ser divididos em três camadas, dependendo da definição escolhida de HDTV ou SDTV, assim:

- Opção 1: os 13 segmentos são utilizados para um programa HDTV ou múltiplos programas SDTV;
- Opção 2: um segmento é utilizado para recepção parcial e 12 segmentos para recepção de um programa HDTV ou múltiplos programas SDTV;
- Opção 3: um segmento é utilizado para recepção parcial, três segmentos para recepção SDTV e nove segmentos para recepção HDTV.

A figura 2.1 mostra a divisão das camadas para 13 segmentos.

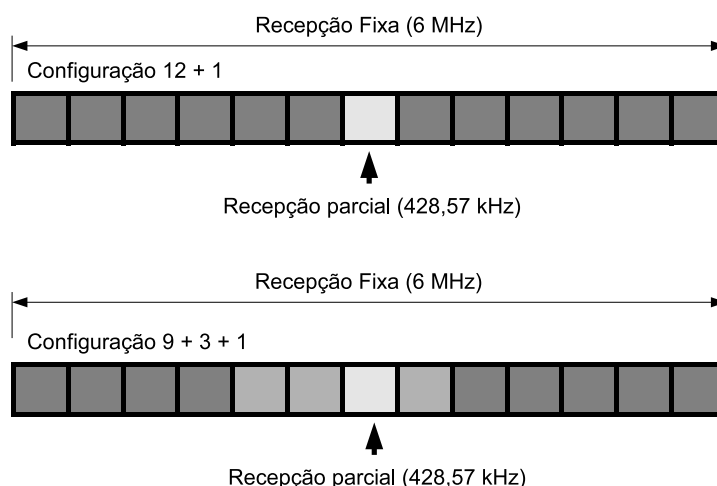


Figura 2.1: Configuração das camadas para os 13 segmentos

Entre os 13 blocos, como mostra a figura 2.1, levando em consideração as opções 2 e 3, o segmento central é destinado para a recepção portátil ou móvel. Dessa forma, o receptor móvel não leva em conta os outros segmentos.

Tanto na transmissão fixa como na transmissão móvel, o vídeo, áudio e os dados de cada programação são empacotados em pacotes chamados de Fluxo de Transporte TS, remultiplexados em Fluxo de Pacotes de Transporte (TSP do inglês: *Transport Stream Packet*). É importante a estrutura do pacote TS, pois deste depende a quantidade de *bytes* que terá os símbolos de entrada e saída do nosso codificador AL-FEC. Também sabendo-se a estrutura do TS, saber-se-á como serão enviados os pacotes de redundância gerados pelo codificador e também poder-se-á saber o modelo de canal visto desde a camada de aplicação.

Seguidamente, depois do empacotamento TSP os dados são submetidos à codificação de canal múltiplo e enviados como um sinal OFDM (OFDM do inglês: *Orthogonal Frequency-Division Multiplexing*). O sistema OFDM pode operar em três modos de multiplexadoras: 2k, 4k e 8k.

O sinal transmitido deve utilizar o entrelaçamento no tempo, para diminuir a taxa de erros no receptor móvel. Mas, quanto maior seja o tempo de entrelaçamento, maior será o tempo de espera depois de uma troca de canal e menor será a quantidade de símbolos

a serem codificados.

## 2.2 *Transport Stream* TS

Um Pacote PES (PES do inglês: *Packetized Elementary Stream*) está constituído de vários ES (ES do inglês: *Elementary Stream*) de vídeo, de áudio e de referência de tempo[7], onde tempo se refere à informação de tempo para a sincronização entre o áudio e o vídeo. Este tempo de sincronização recebe o nome de PCR (PCR do inglês: *Program Clock Reference*).

A estes pacotes PES são acrescentados o SI (SI do inglês: *System Information*) que são informações sobre os dados contidos nos pacotes PES. Assim, o receptor pode distinguir os dados apropriados atrelados a seu programa ou canal de televisão.

Esses programas estão organizados em tabelas, assim a primeira tabela é a PAT (PAT do inglês: *Program Association Table*), e sempre tem o endereço 0000. Esta tabela PAT contém os identificadores PID das tabelas PMT (PMT do inglês: *Program Map Table*), que são os programas transmitidos. Por sua vez, as tabelas PMT contém os identificadores dos PES que compõem o programa.

Por exemplo:

- PAT (PID 0x0000) contém os identificadores PID das tabelas 0x0101 e 0x1fc8;
- PMT 1 (PID 0x0101) = Vídeo PID 0x111, Áudio 1 PID 0x0112, Áudio 2 PID 0x0113, PCR PID 0x0100;
- PMT 2 (PID 0x1fc8) = Vídeo PID 0x211, Áudio 1 PID 0x0212, PCR PID 0x0200.

Como pode-se observar, tem-se dois tipos de programas sendo transmitidos. O primeiro programa (PMT 1) possui um vídeo e dois tipos de áudio que, neste caso, podem ser dois tipos de linguagens diferentes ou dois canais separados de áudio. E no segundo programa (PMT 2) esta sendo transmitido um fluxo de vídeo e um fluxo de áudio. Todos os programas têm uma Identificação de Programa (PID do inglês: *Program Identification*) para tempos de sincronização diferentes.

Além disso, na PAT tem-se a tabela NIT (NIT do inglês: *Network Information Table*), no PID 0x0010. Esta tabela contém informações sobre rede, as multiplexações e taxas de símbolo numa determinada rede.

As Tabelas PMT também apontam à tabela SDT (SDT do inglês: *Service Description Table*), que contém o nome do canal e informações sobre serviços que podem ser multiplexados como parte do PES.

A tabela TOT (TOT do inglês: *Time Offset Table*) provê informação sobre fuso-horário. Isto é usado para definir o tempo local. O valor de PID é 0x0014.

A figura 2.2 mostra como estes pacotes são encapsulados até o TS.

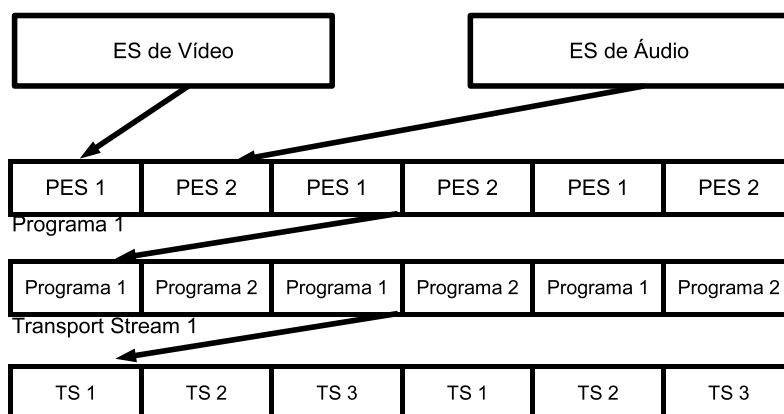


Figura 2.2: Diagrama representativo do encapsulamento do TS

A Figura 2.2, mostra que os pacotes ES são propriamente identificados e empacotados em pacotes PES. Esses pacotes PES, por sua vez, são identificados em tabelas que determinam: que pacote PES pertence a que programa. No final esse pacotes PES propriamente identificados por programas são empacotados em pacotes TS.

Cabe destacar que o pacote PES tem tamanho variável, até 65536 *bytes*. Para manter o tamanho fixo, os pacotes PES são empacotados novamente em pacotes chamados de TS, que são de tamanho fixo de 188 *bytes*, como mostra a Figura 2.2. Assim, vários PES cabem em um pacote TS.

Comumente, os pacotes TS contêm 4 *bytes* de cabeçalho seguidos de 184 *bytes* de

*payload*.

A estrutura TS (veja [8]) possui uma *flag* TEI (TEI do inglês: *Transport Error Indicator*) que indica erro no pacote TS. Assim, se a *flag* for 1, o pacote contém erro de pelo menos 1 *bit*. Se o pacote contém erro, considera-se o pacote TS inteiro errado, já que não se pode saber onde está o erro. Desta forma, observando-se desde a camada de aplicação do SBTVD, é como se o SBTVD se comportasse como um canal BEC (BEC do inglês: *Binary Erasure Channel*).

A estrutura dos pacotes TS e PES segue a norma ABNT (Associação Brasileira de Normas Técnicas) de codificação de vídeo, áudio e multiplexação [8], que se refere a que todos os sinais codificados devem ser multiplexados em pacotes, sinais como: vídeo, áudio, informações para gerenciamento de serviços, segurança de comunicação e registro de uso.

### 2.3 Entrelaçamento no Tempo e na Frequência do SBTVD

O objetivo do entrelaçamento no tempo e na frequência é que cada símbolo experimente um desvanecimento independente. No transmissor do SBTVD, após a combinação dos níveis hierárquicos, segue-se o bloco de entrelaçamento no tempo e na frequência [9]. Este entrelaçamento é obrigatório para a recepção móvel. O entrelaçador no tempo entrelaça as subportadoras entre os símbolos OFDM independentemente para cada camada hierárquica. Nesse caso, os símbolos são atrasados no transmissor e no receptor, ou seja, alguns *frames* chegam atrasados [7]. A profundidade do entrelaçamento é ajustada com o parâmetro  $I$  da equação[10]:

$$\pi(i) = I \times (\text{mod}(5i, N_c)) \quad (2.1)$$

Onde:

- $I$  = Ajuste do tamanho do entrelaçador;
- $i$  = Entrada do entrelaçador temporal;
- $\pi(i)$  = Saída do entrelaçador temporal;

- $N_c$  = Número de portadoras de dados.  $N_c = 96, 192$  ou  $384$  para os modos 2k, 4k e 8k, respectivamente [9].

Segundo Akamine (2004)[10], o atraso do entrelaçador temporal é dado por:

$$A_{ET} = T_F (204 \times N_F - N_{SA}) \quad (2.2)$$

Onde:

- $A_{ET}$  = Atraso do Entrelaçador Temporal;
- $T_F$  = Duração do símbolo OFDM;
- $N_F$  = Número de quadros OFDM atrasados;
- $N_{SA}$  = Número de Símbolos para Ajuste.

Assim, Akamine(2004)[10] apresenta uma tabela de tempo de atraso do entrelaçador temporal, Tabela 2.1. Na tabela, estão detalhados os tempos de atraso para cada modo OFDM com respeito ao ajuste do tamanho do entrelaçador  $I$ .

Tabela 2.1: Atrasos do entrelaçador temporal.

Modo 2k			Modo 4k			Modo 8k		
Tamanho (I)	Número de Símbolos OFDM atrasados	Atraso (ms)	Tamanho (I)	Número de Símbolos OFDM atrasados	Atraso (ms)	Tamanho (I)	Número de Símbolos OFDM atrasados	Atraso (ms)
0	0	0	0	0	0	0	0	0
4	380	95.76	2	190	95.76	1	95	95.76
8	760	191.52	4	380	191.52	2	190	191.52
16	1504	379.00	8	760	379.00	4	380	379.00

É interessante destacar a importância do tempo de atraso do entrelaçador, pois quanto maior o tempo de entrelaçamento, maior é a proteção e menor será a quantidade de símbolos a serem codificados, já que dele depende o tempo de espera depois de uma troca de canal. E um dado importante para esta pesquisa é que o tempo maior a 1 segundo de de troca de canal se torna tedioso para o telespectador [11].



## 2.4 Considerações Finais

Neste capítulo, foi vista uma breve introdução do SBTVD focado à recepção móvel e à importância do pacote TS para a estrutura de codificação AL-FEC. Mostrou-se também que, com a estrutura do TS, o SBTVD é visto como um canal com apagamento. Por último, destacou-se a importância do tempo de entrelaçamento para saber a quantidade de símbolos a serem codificados pelo AL-FEC para que o tempo de troca de canal não se torne tedioso. No próximo capítulo, detalhar-se-ão os códigos a serem utilizados na camada de aplicação, neste caso os Códigos fontanais.

## CAPÍTULO 3

### CÓDIGOS FONTANAIS

#### 3.1 Introdução

A proposta da correção de erros na camada de aplicação é de recuperar perdas de pacotes, ou seja, o codificador gerará pacotes redundantes a partir dos pacotes originais para a recuperação dos pacotes errados, diferente da camada física onde o codificador gerará *bits* redundantes a partir dos pacotes originais para a recuperação dos *bits* errados.

O esquema relativamente recente a ser apresentado neste capítulo é do tipo fontanal, analogamente ao do tipo fonte de água, ou fontanal, na qual a fonte, ou codificador, produz um número ilimitado de gotas (códigos ditos sem taxa), onde as gotas são associadas a pacotes. Para que o código seja chamado de “ótimo”, o mesmo precisa decodificar um balde de gotas de água de tamanho original. Os códigos que serão apresentados são códigos “quase ótimos”, que para poder recuperar os pacotes transmitidos, ou codificados, requerem pacotes com o tamanho original mais um pequena quantidade (que chamaremos de redundância, ou em inglês: *overhead*).

Os códigos a serem revisados neste capítulo são: os Códigos LT (LT do inglês: Luby Transform) apresentados por Michael Luby no ano 2002[12] e os Códigos Raptor, apresentados por Shokrollahi (2006)[13], que são da família dos Códigos fontanais.

#### 3.2 Códigos LT

Os Códigos LT, ou Códigos Transformada de Luby, inicialmente apresentado por Michael Luby no ano 2002[12], são da família dos Códigos fontanais inicialmente desenvolvidos para canais BEC. Os LT são códigos sem taxa, ou de taxa versátil, porque o número de símbolos (bloco de *bits*) codificados pode tender a infinito, são de complexidade baixa já que as operações nas que se baseiam são do tipo XOR.

Os Códigos LT são sistemas que produzem a partir de um bloco:

$$S = (S_1, S_2, S_3, \dots, S_k)$$

com  $k$  símbolos de entrada, um bloco:

$$t = (t_1, t_2, t_3, \dots, t_n)$$

com  $n$  símbolos codificados.

Com estes símbolos, podem-se obter os  $k$  símbolos originais com uma probabilidade de  $(1 - \delta)$ , a partir de quaisquer  $k + \left(\sqrt{k} \ln^2 \left(\frac{k}{\delta}\right)\right)$  símbolos codificados, com um limite de  $O\left(k \ln \frac{k}{\delta}\right)$  operações por símbolo [12].

Os Códigos LT são “quase ótimos” em canais com apagamento. Isto quer dizer que recuperam o bloco inicial a partir de quaisquer  $n = (1 + \text{redundância})k$  símbolos codificados, onde a  $0 < \text{redundância} < 1$  [14].

### 3.2.1 Codificação LT

A codificação é conceptualmente bem simples de descrever. Cada símbolo codificado é gerado como a seguir:

1. Um grau  $d$  é aleatoriamente escolhido de acordo com a distribuição  $\mu(d)$  que será apresentada nas seções seguintes;
2. Escolhem-se aleatoriamente  $d$  distintos *bits* de entrada;
3. O *bit* codificado é a soma XOR desses  $d$  *bits*.

### 3.2.2 Codificador LT como um Código em Grafo

O codificador LT pode também ser representado por um grafo bi-particionado como mostra a Figura 3.1 [14]. A figura ilustra um grafo em que o conjunto de nós é particionado em dois conjuntos, sendo o primeiro um conjunto  $S = \{S_1, S_2, \dots, S_i, \dots, S_k\}$  de nós de mensagem, ou nós de entrada, onde no exemplo da figura  $k = 3$ , e o segundo um conjunto  $t = \{t_1, t_2, \dots, t_j, \dots, t_n\}$  de nós de verificação, ou nós de saída, onde no exemplo

da figura  $n = 4$ . A cada nó de entrada associa-se um símbolo de entrada  $S_i$ , que será identificado, indistintamente, como nó  $S_i$ , e a cada nó de saída corresponde um símbolo codificado  $t_j$ , que será identificado como nó  $t_j$ .

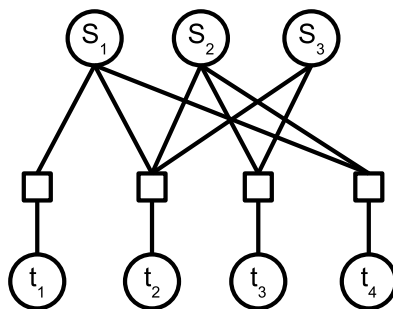


Figura 3.1: Grafo de codificação

Na Figura 3.1, apresenta-se o gráfico onde a mensagem de entrada é colocada acima em forma de círculos e a palavra codificada é colocada abaixo, este grafo também é chamado de “Grafo de Tanner”.

Na Tabela 3.1, pode-se ver o caso particular de codificação a que faz referência à figura 3.1, onde  $n = 4$  e  $k = 3$ . A tabela mostra que para cada símbolo codificado  $t_j$  é designado um número inteiro, tal que  $d_j \in \mathbb{Z}_k$ , que é chamado de grau  $d_j$ . O grau  $d_j$  é obtido a partir de uma distribuição de probabilidades  $\mu(d)$ , que será explicada mais adiante.

Chamam-se vizinhos ao o conjunto de símbolos de entrada  $\nu_j = \{S_{i_1}, \dots, S_{i_{d_j}}\}$ , onde os símbolos de entrada são escolhidos aleatoriamente com quantidade dada pelo grau  $d_j$ . Assim, cada  $t_j$  é a soma módulo 2, de todos os símbolos de entrada contidos no vizinho.

Na Tabela 3.1, pode-se ver um exemplo de codificação. Para uma mensagem  $S_1 S_2 S_3 = 101$ , tem-se o símbolo codificado  $t_j$  e o grau.

Tabela 3.1: Representação do grafo biparticionado

Símbolo	Grau $d_j$	Vizinhos	Valor
$t_1$	1	$S_1$	$S_1 = 1$
$t_2$	3	$S_1 S_2 S_3$	$S_1 \oplus S_2 \oplus S_3 = 0$
$t_3$	2	$S_2 S_3$	$S_2 \oplus S_3 = 1$
$t_4$	2	$S_1 S_2$	$S_1 \oplus S_2 = 1$

### 3.2.3 Decodificação LT

Para decodificar os  $k$  símbolos de entrada, supõe-se que o decodificador conhece o grafo do codificador. Neste processo de decodificação, utiliza-se a mesma notação: os símbolos codificados  $t_j$  formarão os nós de saída e o símbolos de entrada  $S_k$  formarão os nós de entrada. O algoritmo de decodificação de Códigos LT é descrito a seguir:

1. Encontra-se o nó  $t_j$  que esteja conectado a somente um nó de entrada  $S_k$ . (Se este não existir, a decodificação pára neste ponto e falha);
  - (a) Faz-se  $S_k = t_j$ ;
  - (b) Acrescenta-se  $S_k$  a todos os nós  $t_j$  que estão conectados aos  $S_k$ , com a operação XOR;
  - (c) Remove-se todas as conexões que chegam ao símbolo de entrada  $S_k$ .
2. Repete-se o passo (1), até que todos os  $S_k$  símbolos de entrada sejam determinados.

A Figura 3.2 mostra o processo de decodificação para a mensagem codificada anteriormente (Figura 3.1). Na Figura 3.2, no passo 1, encontra-se o nó  $t_1$  que está conectado a somente um nó de entrada  $S_1$  e se faz  $S_1 = t_1$ . No passo 2, acrescenta-se  $S_1$  aos nós  $t_2$  e  $t_4$  que estão conectados ao  $S_1$  e se removem todas as conexões que chegam ao símbolo de entrada  $S_1$ . No passo 3, encontra-se o nó  $t_4$  que está conectado a somente um nó de entrada  $S_2$  e se faz  $S_2 = t_4$ . No passo 4, acrescenta-se  $S_2$  aos nós  $t_2$  e  $t_3$  que estão conectados ao  $S_2$  e se removem todas as conexões que chegam ao símbolo de entrada  $S_2$ . No passo 5, existem os  $t_2$  e  $t_3$ , que têm o mesmo símbolo, conectados somente a um nó  $S_3$ , como os nós  $t_j$  têm o mesmo símbolo, então se faz  $S_3 = t_2 = t_3$ . Por fim, no passo 6, removem-se todas as conexões que chegam ao símbolo de entrada  $S_3$  e tem-se assim o resultado da decodificação.

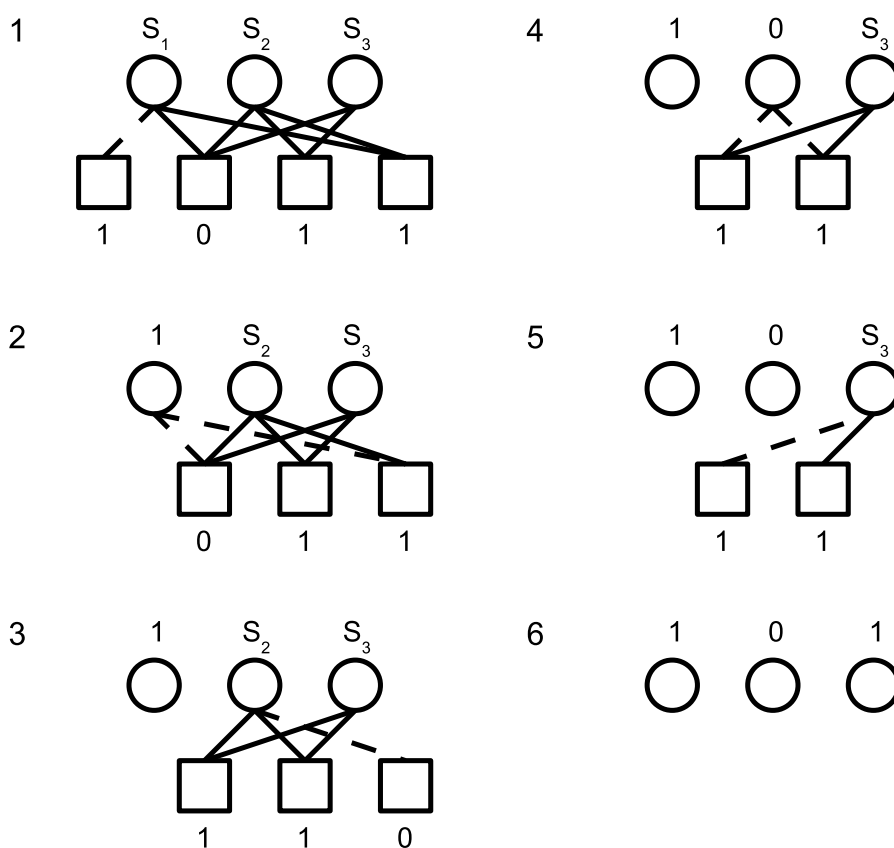


Figura 3.2: Processo de decodificação

### 3.2.4 Distribuição de Graus

A distribuição de graus é muito importante para a codificação LT, dela depende manter um número de símbolos codificados  $n = k + (\varepsilon \times k)$  bem próximo ao número de símbolos originais, onde  $(\varepsilon \times k)$  é o número de símbolos de paridade que é acrescentado aos  $k$  símbolos originais para poder depois corrigir os erros, se for o caso. Na Equação,  $\varepsilon$  representa um número de 0 a 1.

A distribuição deve conter graus  $d_j \approx k$ , sendo  $k$  o tamanho dos símbolos originais e  $d_j$  a quantidade de símbolos originais escolhidos aleatoriamente. Essa premissa assegura que a maioria dos símbolos estejam conectados. Isso implica na diminuição de erros na decodificação causados pela falta de combinação de símbolos.

É importante também, que a distribuição de graus contenha muitos pacotes de graus

$d_j$ , bem pequenos, para começar a decodificação e manter baixo o número de operações XOR e o esforço computacional.

A seguir, apresentaremos a Distribuição Sóliton Robusta apresentada inicialmente por Luby [12], que é um aperfeiçoamento da Sóliton Ideal.

## Distribuição Sóliton Robusta

Apresentada por Luby, a Distribuição Sóliton Ideal  $\rho(d)$  para um tamanho de mensagem é definida como [12] :

$$\rho(d) = \begin{cases} \frac{1}{k}, & \text{para } d=1 \\ \frac{1}{d(d-1)}, & \text{para } d=2, 3, \dots, k. \end{cases} \quad (3.1)$$

Na prática, esta distribuição tem desempenho muito baixo, porque nem sempre consegue-se nós de grau 1, fazendo com que o processo de decodificação termine sem sucesso. Além disso, alguns símbolos de entrada não receberam nenhuma conexão.

Com a introdução da Distribuição Sóliton Robusta, são acrescentados dois parâmetros,  $c$  e  $\delta$ . Assim, consegue-se que o número esperado de nós de grau 1 seja aproximado a  $R \equiv c \ln\left(\frac{k}{\delta}\right) \sqrt{k}$ . O parâmetro  $\delta$  é um limitante da probabilidade de falha do processo de decodificação, depois de que um certo número de pacotes sejam recebidos [12] [15] . O parâmetro  $c$  é uma constante de ordem 1, mas normalmente se apresenta como um número menor que 1 e maior do que 0.

Com esses parâmetros, define-se a seguinte função:

$$\tau(d) = \begin{cases} \frac{R}{dk}, & \text{para } d = 1, 2, 3 \dots \left(\frac{k}{R} - 1\right) \\ \frac{R}{k} \log\left(\frac{R}{\delta}\right), & \text{para } d = \frac{k}{R}, \\ 0, & \text{para } d > \frac{k}{R} \end{cases} \quad (3.2)$$

Somando as distribuições e normalizando, obtém-se:

$$\mu(d) = \frac{\rho(d) + \tau(d)}{\beta}, \text{ para } 1 \leq d \leq k \quad (3.3)$$

onde  $\beta = \sum_{d=1}^k (\rho(d) + \tau(d))$  é uma constante de normalização.

A Equação 3.3 é conhecida como distribuição Sóliton Robusta.

A Figura 3.3 mostra a distribuição Sóliton Robusta para o caso  $k = 10000$ ,  $c = 0,2$  e  $\delta = 0,1$ .

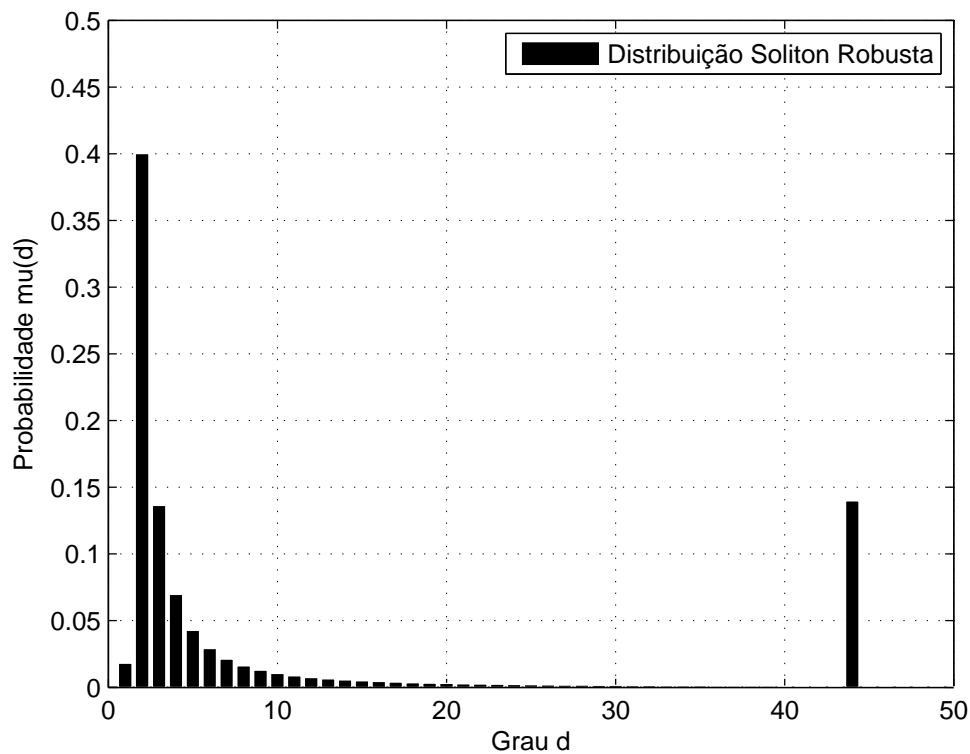


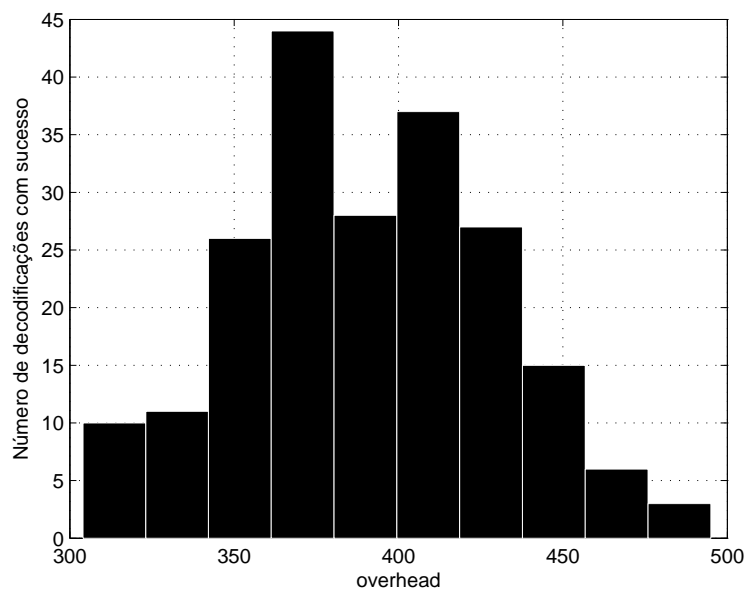
Figura 3.3: Distribuição Sóliton Robusta para  $k = 10000$ ,  $c = 0,2$  e  $\delta = 0,1$

Paiba (2009) [16] mostrou que os parâmetros de  $c = 0,03$  e  $\delta = 0,1$  geram bons Códigos LT. Considera-se um bom código LT aquele que é capaz de recuperar a mensagem original a partir de um número de símbolos de saída  $n$ ; sendo que esse  $n$  é um pouco maior que o número de símbolos de entrada  $k$  que formam a mensagem original e  $n = k + (k \times 0,1)$ .

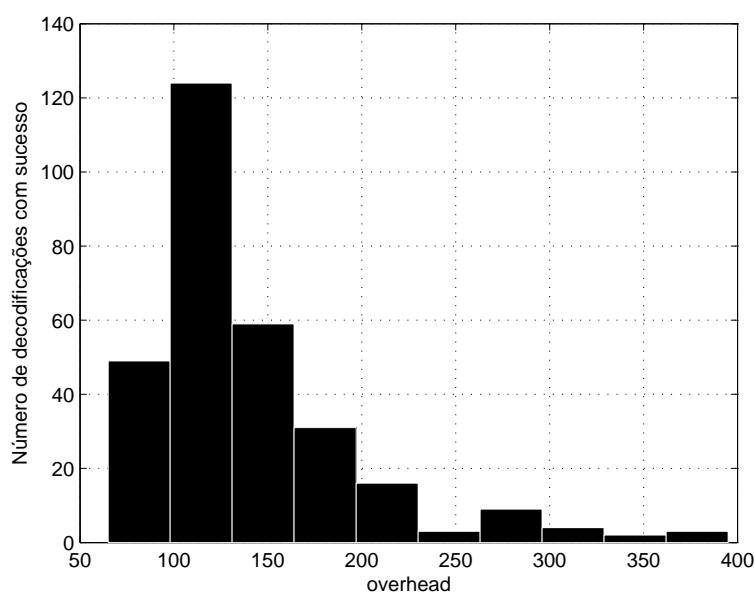
As figuras 3.4a e 3.4b mostram o histograma do número de símbolos de *overhead* necessários para a decodificação com sucesso. Entende-se *overhead* na Figura 3.4 como a quantidade de símbolos além dos  $k$  símbolos necessários para a decodificação. A Figura 3.4b mostra os símbolos de *overhead* necessários para os valores de  $c = 0,03$  e  $\delta = 0,1$ , já a Figura 3.4a, para os valores de  $c = 0,2$  e  $\delta = 0,1$ . Como podemos observar, a quantidade média de símbolos de *overhead* necessários para a decodificação com sucesso é menor para



os valores de  $c=0,03$  e  $\delta = 0,1$ .



(a) Histograma do número de símbolos de *overhead* ( $\varepsilon \times k$ ) necessários para uma decodificação com sucesso para  $k = 1000$ ,  $c = 0,2$  e  $\delta = 0,1$



(b) Histograma do número de símbolos de *overhead* ( $\varepsilon \times k$ ) necessários para uma decodificação com sucesso para  $k = 1000$ ,  $c = 0,03$  e  $\delta = 0,1$

Figura 3.4: Histograma do número de símbolos de *overhead* necessários para uma decodificação com sucesso para  $k = 1000$  e distintos valores de  $c$  e  $\delta$

Nesta seção, foi apresentada a codificação LT, que é da família dos Códigos fontanais, dita sem taxa. Foi também apresentada a distribuição inicialmente proposta por Luby

[12]. E na seção seguinte, serão apresentados os Códigos Raptor que são uma evolução dos Códigos fontanais.

O Código Raptor foi aprovado pela 3GPP (3GPP do inglês: *3rd Generation Partnership Project*) para a utilização em terminais que são parte do MBMS (MBMS do inglês: *Multimedia Broadcast Multicast Services*) [17], como também o DVB-H optou pelo código na correção de erros na camada de aplicação.

### 3.3 Códigos de Raptor

#### 3.3.1 Códigos de Raptor Não-Sistemáticos

Os Códigos de Raptor foram inicialmente introduzidos por Shokrollahi [13], como Códigos fontanais. Nos Códigos Raptor, diferentemente dos Códigos LT, existe uma etapa de pré-codificação, na qual é introduzido um código de bloco sistemático de comprimento fixo. Neste caso, a fim de simplificar o estudo, no pré-codificador é utilizado o Código LDPC.

Como mostra a Figura 3.5, a partir de um bloco  $\mathbf{S} = (S_1, S_2, S_3, \dots, S_k)$  com  $k$  símbolos de entrada, a matriz geradora da pré-codificação  $\mathbf{G}_P$ , de tamanho  $[m \times k]$ , produz um vetor  $\mathbf{F} = (F_1, F_2, F_3, \dots, F_L)$ , onde  $L = k + m$  símbolos pré-codificados, sendo que  $m$  é o número de símbolos de paridade gerado pelo código LDPC, que equivale ao vetor  $\mathbf{D}_P = (D_{P1}, D_{P2}, D_{P3}, \dots, D_{Pm})$ .

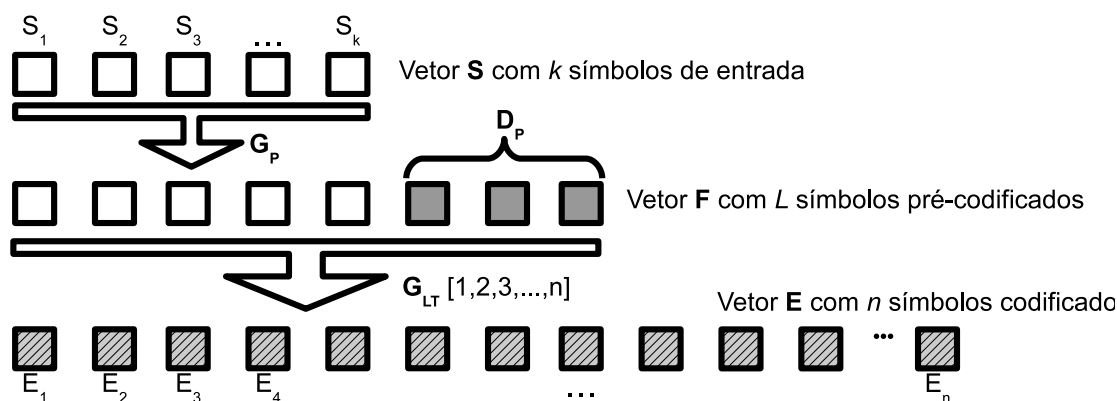


Figura 3.5: Estágio de codificação do Código Raptor

Os símbolos de paridade  $\mathbf{D}_P$  são obtidos da seguinte forma:

$$\mathbf{D}_P = \mathbf{G}_P \cdot \mathbf{S} \quad (3.4)$$

O conjunto de símbolos pré-codificados  $\mathbf{F}$  é dado por  $[\mathbf{S}^T \mathbf{D}_P^T]^T$ , de dimensão  $[(k + m) \times 1]$ . Estes símbolos pré-codificados servem como símbolos de entrada para o codificador LT, que é o mesmo Código LT que vimos no capítulo 3.2 com a diferença na distribuição de graus, o que será visto mais adiante. Então, os símbolos codificados no vetor  $\mathbf{E}_{[1:n]}$ , de dimensão  $[n \times 1]$ , são obtidos como segue:

$$\mathbf{E}_{[1:n]} = \mathbf{G}_{LT}(1,2,\dots,n) \cdot [\mathbf{S}^T \mathbf{D}_P^T]^T \quad (3.5)$$

Na Equação 3.5, a matriz  $\mathbf{G}_{LT}$  tem dimensão  $[n \times (k + m)]$ .

As Equações 3.4 e 3.5 podem ser representados como:

$$\mathbf{A}_{[1,2,\dots,n]} \cdot \begin{pmatrix} \mathbf{S} \\ \mathbf{D}_P \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{E}_{[1:n]} \end{pmatrix} \quad (3.6)$$

Na Equação 3.6, o “ $\mathbf{0}$ ” é um vetor que tem dimensão  $[m \times 1]$ . A matriz  $\mathbf{A}$  tem dimensão  $[(m + n) \times (k + m)]$  e é conhecida como “*Matriz de Codificação A*”. É definida como:

$$\mathbf{A}(1,2,\dots,n) \triangleq \begin{bmatrix} \mathbf{G}_P \mathbf{I}_m \\ \mathbf{G}_{LT}(1,\dots,n) \end{bmatrix} \quad (3.7)$$

Da Equação 3.7,  $\mathbf{I}_m$  é a matriz identidade de dimensão  $[m \times m]$ . A matriz de codificação  $\mathbf{A}$  não é uma matriz geradora e sim uma matriz que fornece um conjunto de restrições tanto para a pré-codificação, quanto para a codificação LT.

Na decodificação do Código Raptor, assumindo que o canal tem perdas de pacotes, é recebido um subconjunto de símbolos identificado pelo vetor  $\mathbf{i} = (i_1, i_2, \dots, i_r)$ , onde  $r \leq n$ , a decodificação é realizada baseada na matriz de decodificação  $\mathbf{A}_{[i_1, i_2, \dots, i_r]}$ , definida como:

$$\mathbf{A}(i_1, i_2, \dots, i_r) \triangleq \begin{bmatrix} \mathbf{G}_P \mathbf{I}_m \\ \mathbf{G}_{LT}(i_1, \dots, i_r) \end{bmatrix} \quad (3.8)$$

A matriz de decodificação da Equação 3.8 é uma generalização da matriz de codificação da Equação 3.7, porque os índices não são mais consecutivos, mas somente os índices recebidos são levados em conta.

Com a matriz de decodificação  $\mathbf{A}$ , Equação 3.8, o processo de decodificação é resolvido com a Equação 3.9.

$$\mathbf{A}(i_1, i_2, \dots, i_r) \cdot \mathbf{F} = (\mathbf{0}^T \ E_{i_1}^T \ E_{i_2}^T \ \dots \ E_{i_r}^T) \quad (3.9)$$

Na Equação 3.9, o “ $\mathbf{0}$ ” é um vetor que tem dimensão  $[m \times 1]$ . Percebe-se que os primeiros  $k$  símbolos do vetor  $\mathbf{F}$  correspondem aos símbolos intermediários, segundo a Equação 3.10.

$$\mathbf{F}_{[1:k]} = \mathbf{S} \quad (3.10)$$

Apesar dos Códigos Raptor não-sistemáticos terem um bom desempenho, precisam-se de códigos sistemáticos para o uso no SBTVD, visando não afetar o sistema já implementado.

### 3.3.2 Codificação Raptor Sistemática

Para obter-se um código sistemático, é necessário que se encontre um mapeamento apropriado das palavras de informação às palavras código, tais que os primeiros  $k$  símbolos da palavra código  $\mathbf{E}_i$  correspondam aos primeiros símbolos fonte  $\mathbf{C}_i$  (vide a figura 3.6), isto é:

$$\mathbf{E}_i = \mathbf{C}_i \ \forall i = 1, 2, \dots, k \quad (3.11)$$

Assume-se que a matriz geradora de um Código Raptor não-sistemático para os primeiros  $k$  símbolos é denotada pela matriz  $\mathbf{G}_k$  de dimensão  $[k \times k]$ , tal que  $\mathbf{E}_{1:k} = \mathbf{G}_k \cdot \mathbf{S}$ . Então, para garantir a Equação 3.11, 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.

Aproveitando o fato que  $\mathbf{E}_{[1:k]} = \mathbf{C}$ , isto é, a Equação 3.11 e as equações 3.4 e 3.5, obtemos:

$$\mathbf{G}_k = \mathbf{G}_{LT}(1,2,\dots,k) \cdot \begin{bmatrix} \mathbf{I}_k \\ \mathbf{G}_P \end{bmatrix}. \quad (3.12)$$

Depois de resolvida a Equação 3.12, os símbolos fonte podem facilmente ser transformados em símbolos intermediários  $\mathbf{S}$ , como  $\mathbf{S} = \mathbf{G}_k^{(-1)} \cdot \mathbf{C}$ .

Na Figura 3.6, está representada a estrutura dos Códigos Raptor Sistemáticos, que é a concatenação de  $\mathbf{G}_k^{(-1)}$ , o pré-codificador  $\mathbf{G}_P$  e o codificador  $\mathbf{G}_{LT}$ .

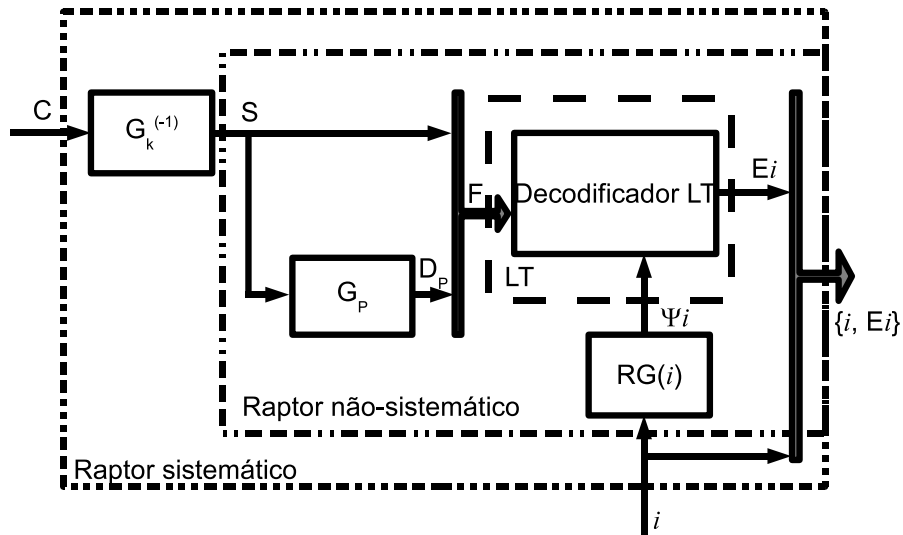


Figura 3.6: Diagrama em bloco da Codificação Raptor Sistemática Conceptual

Entretanto, a implementação da inversa de  $\mathbf{G}_k$  é muito ineficiente. Uma forma diferente de implementação foi apresentada em [17]. Sugere-se que a codificação pode ser feita usando uma “*code constraint matrix*” específica, ou seja, uma matriz de codificação conforme definida em 3.7 para  $n = k$ , isto é  $\mathbf{A}_{[1,\dots,n=k]}$ . Desta forma, os símbolos intermediários  $\mathbf{S}$  não necessitam ser calculados explicitamente e os símbolos pré-codificados  $\mathbf{F}$  são obtidos resolvendo-se a Equação:

$$\mathbf{A}_{[1,\dots,k]} \cdot \mathbf{F} \equiv (\mathbf{0}^T \mathbf{C}^T)^T \quad (3.13)$$

Os símbolos intermediários  $\mathbf{S}$  são obtidos inerentemente a partir de  $\mathbf{F}$ , conforme a Equação 3.10. Nota-se também que as restrições na Equação 3.13 são equivalentes ao processo de decodificação Raptor, conforme introduzido na Equação 3.6. Os símbolos resultantes  $\mathbf{S}$  podem agora ser utilizados como símbolos de entrada para o código Raptor não-sistemático.

Nas seguintes subseções, será apresentada a construção da matriz de codificação  $\mathbf{A}$ , assim, precisa-se da construção da matriz de paridade  $\mathbf{G}_P$  e da matriz LT  $\mathbf{G}_{LT}$ .

### 3.3.3 Matriz de Pré-Codificação

Nesta seção, serão descritas de maneira breve as características principais da matriz de paridade  $\mathbf{G}_P$  dos Códigos LDPC.

Normalmente, a matriz de paridade dos Códigos LDPC é chamada de matriz  $\mathbf{H}$ , mas neste caso, para fins de entendimento atrelado à explicação da construção da matriz de codificação  $\mathbf{A}$  do Código Raptor deste trabalho, chamaremos a matriz  $\mathbf{H}$  de matriz  $\mathbf{G}_P$ .

Os Códigos LDPC, são códigos lineares de bloco, com matriz de paridade  $\mathbf{G}_P$ , neste trabalho utilizaremos o tamanho  $[(L - k) \times L]$ , onde  $L$  é o número de símbolos pré-codificados. A matriz  $\mathbf{G}_P$  é uma matriz esparsa, ou seja, números de “1” muito menores que o número total de valores na matriz.

Designa-se  $w_c$  como peso das colunas, ou seja, como a quantidade de 1s presentes nas colunas. Assim também, designa-se  $w_r$  como peso das linhas, ou seja, como a quantidade de 1s presentes nas linhas.

Quanto à construção, as matrizes LDPC podem ser regulares ou irregulares. É chamada de  $(w_c, w_r)$  regular se todas as colunas têm o mesmo número de 1s; e, independentemente das colunas, se todas as filas tem também o mesmo número de 1s na matriz  $\mathbf{G}_P$ . É dita irregular se o número de 1s, ou o peso, é diferente para cada coluna ou linha da matriz  $\mathbf{G}_P$ .

Para uma matriz irregular, designa-se a fração de colunas de peso  $i$  como  $v_i$  e a fração de linhas de peso  $i$ , como  $h_i$ . Assim, o par  $v, h$ , onde  $v = [v_1, v_2, v_3, \dots]$  e  $h = [h_1, h_2, h_3, \dots]$ , é chamado de distribuição de grau do código.

Normalmente, as matrizes LDPC são representadas em sua forma gráfica pelo “Grafo de Tanner”. Este é um grafo bipartido que contém os nós de paridade e os nós de variável.

Uma caracterização alternativa da distribuição de grau, vista da perspectiva do grau de conexões no Grafo de Tanner é a distribuição de grau das conexões. Tal que, a fração de 1s que tem peso  $i$  nas colunas é denotada de  $\lambda_i$  e a fração de 1s que tem peso  $i$  nas filas é denotada como  $\rho_i$ . Então,

$$\lambda(x) = \sum_i \lambda_i x^{i-1} \quad (3.14)$$

$$\rho(x) = \sum_i \rho_i x^{i-1} \quad (3.15)$$

Para a construção da matriz de paridade a partir das distribuições de grau ( $\lambda$  e  $\rho$ ), neste trabalho, usa-se o algoritmo PEG (PEG do inglês: *Progressive Edge Growth*) que será detalhado no Capítulo 4 - Metodologia.

Tendo as distribuições de grau ( $\lambda$  e  $\rho$ ), precisa-se da taxa para a construção da matriz de paridade  $\mathbf{G}_P$ , onde a taxa  $R$  referente ao pré-código para o Código Raptor é dada por:

$$R = \frac{\left(1 + \frac{\varepsilon_{Raptor}}{2}\right)}{\left(1 + \varepsilon_{Raptor}\right)} \quad (3.16)$$

e  $\varepsilon_{Raptor}$  é o *overhead* do Código Raptor.

Dado a taxa  $R$  da pre-codificação precisa-se do *overhead* do Código Raptor. Assim, na seguinte subseção será abordado o *overhead* do Código Raptor.

### 3.3.4 *Overhead* do Código Raptor

O *overhead* total do Código Raptor,  $\varepsilon_{Raptor}$ , depende dos *overheads* do pré-codificador e do codificador *wLT* [13]

$$(1 + \varepsilon_{Raptor}) = (1 + \varepsilon_{LDPC}) \cdot (1 + \varepsilon_{wLT}) \quad (3.17)$$

Onde: ( $wLT$ ) é a “distribuição de Shokrollahi” ou “código LT enfraquecido”, o  $\varepsilon_{wLT}$  é o *overhead* do código LT enfraquecido e  $\varepsilon_{LDPC}$  é o *overhead* da pre-codificação.

Assim, Shokrollahi sugere que  $\varepsilon_{wLT} = 0,5 \varepsilon_{Raptor}$ . Então, a Equação 3.17 é reduzida a:

$$\varepsilon_{Raptor} = \frac{2 \varepsilon_{LDPC}}{1 - \varepsilon_{LDPC}} \quad (3.18)$$

Até aqui, pode-se construir a matriz  $\mathbf{G}_P$  dada uma taxa  $R$  e um *overhead* de Raptor  $\varepsilon_{Raptor}$ , para a matriz geradora  $\mathbf{A}_{[1, \dots, n=k]}$ . Ainda precisa-se da distribuição de graus para matriz de codificação LT  $\mathbf{G}_{LT}$  que será apresentada na seguinte subseção.

### 3.3.5 Código Interno LT

A distribuição de graus para a matriz LT utilizada no Raptor foi proposta por Shokrollahi [13], onde sugere uma distribuição de graus de no máximo  $D + 1$  que é menor a  $k$ .

Sendo  $\varepsilon_{Raptor}$  o *overhead* do código Raptor, número real maior do que zero, é definida como distribuição de Shokrollahi a Equação 3.19.

$$\Omega_D(x) = \frac{1}{\mu + 1} \left( \mu x + \frac{x^2}{1 \cdot 2} + \frac{x^3}{2 \cdot 3} + \dots + \frac{x^D}{(D-1) \cdot D} + \frac{x^{D+1}}{D} \right) \quad (3.19)$$

Onde, na Equação  $\mu = (\varepsilon_{Raptor}/2) + (\varepsilon_{Raptor}/2)^2$  e  $D = \lceil 4(1 + \varepsilon_{Raptor}) / \varepsilon_{Raptor} \rceil$

Com a distribuição de graus, pode-se construir a matriz  $\mathbf{G}_{LT}$  do código interno LT. Assim, conclui-se a construção da matriz de codificação  $\mathbf{A}_{[1, \dots, n=k]}$  e, conseqüentemente, tem-se a construção da matriz de decodificação  $\mathbf{A}_{[i_1, i_2, \dots, i_r]}$ .

## 3.4 Considerações Finais

Neste capítulo, foram apresentados dois tipos de Códigos fontanais: os Códigos LT e os Códigos Raptor, tanto quanto a codificação e decodificação dos mesmos. No próximo



capítulo, será apresentada a metodologia para chegar-se ao objetivo desta pesquisa.

## CAPÍTULO 4

### METODOLOGIA

#### 4.1 Introdução

O processo de codificação na camada de aplicação deve ser transparente. Para que isto seja feito, o vídeo e o áudio não devem ser alterados.

A Figura 4.1 mostra novamente a arquitetura proposta. Neste caso, a metodologia será um pouco mais detalhada, segundo os estudos feitos até aqui. Como pode-se observar, o áudio e o vídeo são empacotados num fluxo elementar, ou PES, e processados pelo AL-FEC que gera o fluxo elementar de correção de erro FEC ES. Estes pacotes ES's são multiplexados com o processo TS e propagados pelo SBTVD.

A estrutura TS (veja [8]) possui uma *flag* TEI (do inglês: *Transport Error Indicator*) que indica erro no pacote TS. Assim, se a *flag* for 1, o pacote contém erro de pelo menos 1 *bit*. Se o pacote contém erro, considera-se o pacote TS inteiro errado, já que não se pode saber onde está o erro. Este procedimento se assemelha ao canal BEC.

Receptores que não têm a arquitetura proposta, ilustração do centro da Figura 4.1, descartariam os pacotes FEC ES.

Receptores que possuem a arquitetura proposta, última ilustração da Figura 4.1, utilizariam o FEC ES para reconstruir áudio e vídeo com erros.

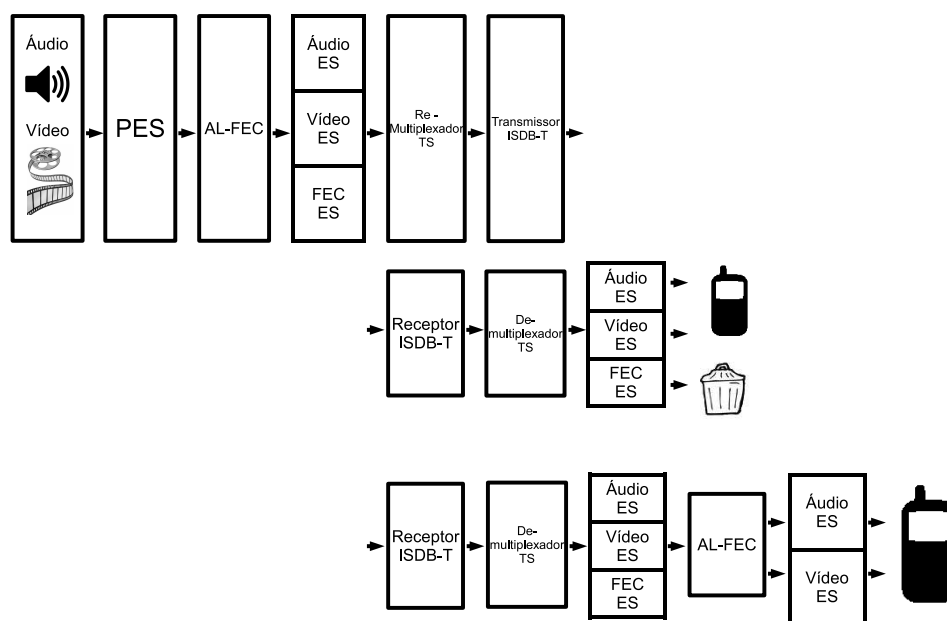


Figura 4.1: Arquitetura do sistema proposto

Os pacotes de erros FEC ES devem ser empacotados em pacotes de transporte TS, com um PID diferente ao de áudio, de vídeo ou dos utilizados, para que no receptor os aparelhos móveis que não reconheçam esse PID facilmente o descartem.

A utilização dos diferentes PID pelos fluxos de FEC ES gerados pelo AL-FEC faz que o número de serviços utilizados seja reduzido no transporte de TS.

A maior vantagem de utilizar-se o AL-FEC é aumentar o desempenho para recepção móvel.

Neste trabalho, para simulação do SBTVD, utiliza-se o simulador SBTVD apresentado por Schwartz [18] (vide Capítulo 4.4). Devido ao fato que a simulação na ferramenta de Schwartz consome muito tempo para cada simulação, utiliza-se o modelo de Cadeia de Markov de Dois Estados para simular o comportamento do simulador SBTVD de Schwartz.

## 4.2 Códigos Fontanais

### 4.2.1 Pré-código LDPC para Código Raptor

O algoritmo utilizado para a geração da matriz de paridade  $\mathbf{G}_P$ , da pré-codificação LDPC, é o algoritmo PEG que foi proposto originalmente em [19]. Este algoritmo cria a matriz de paridade  $\mathbf{G}_P$  para LDPC, dada a função de distribuição de grau  $\lambda$  escolhida.

Dado um número de nós de variáveis, um número de nós de verificação e com a sequência de nós de variáveis  $\lambda$ , o objetivo do algoritmo é inserir uma aresta por vez no Grafo de Tanner e cada aresta é adicionada de forma a maximizar o *girth* local. Lembrando que o *girth* é o ciclo de menor tamanho e, lembrando também, que o ciclo é a sequência de nós conectados que começa e termina no mesmo nó e que não contenha o mesmo nó mais de uma vez.

Em [13], é mencionado que para o pré-código Raptor pode ser utilizada uma distribuição de grau dada por  $\lambda = (2x + 3x^2) / 5$ . Esta distribuição de grau é também utilizada nos artigos [20], [21], [22], entre outros.

### 4.2.2 Distribuição de Graus do LT para o Código Raptor

Na Figura 4.2, comparam-se as distribuições de graus dadas pela Equação 3.19 com a tabela de distribuição de grau apresentada por Shokrollahi [13], para  $k = 65536$ , onde  $k$  é o número de símbolos de entrada, conforme a Tabela 4.1. Pode-se observar que a distribuição dada pela Equação 3.19 tem menor desempenho que a da tabela apresentada no artigo [13], de pelo menos 0,2 dB para uma BER de  $1 \times 10^{-7}$ . A curva corresponde aos parâmetros: Camada Central, Modo 3, Modulação QPSK, Taxa do Código Convolutacional 1/2, Intervalo de Guarda 1/16, Comprimento do *Interleaving* 4, Canal Brasil A  $f_d = 10 H_z$  método *Restricted Jakes*.

Tabela 4.1: Graus de distribuição para o código interno LT para diferentes valores de  $k$ 

$kmax$	65536	80000	100000	120000
$\Omega_1$	0,007969	0,007544	0,006495	0,004807
$\Omega_2$	0,493570	0,493610	0,495044	0,496472
$\Omega_3$	0,166220	0,166458	0,168010	0,166912
$\Omega_4$	0,072646	0,071243	0,067900	0,073374
$\Omega_5$	0,082558	0,084913	0,089209	0,082206
$\Omega_8$	0,056058	0,049633	0,041731	0,057471
$\Omega_9$	0,037229	0,043365	0,050162	0,035951
$\Omega_{18}$	0	0	0	0,001167
$\Omega_{19}$	0,055590	0,045231	0,038837	0,054305
$\Omega_{20}$	0	0,010157	0,015537	0
$\Omega_{65}$	0,025023	0	0	0,018235
$\Omega_{66}$	0,003135	0,010479	0,016298	0,009100
$\Omega_{67}$	0	0,017365	0,010777	0

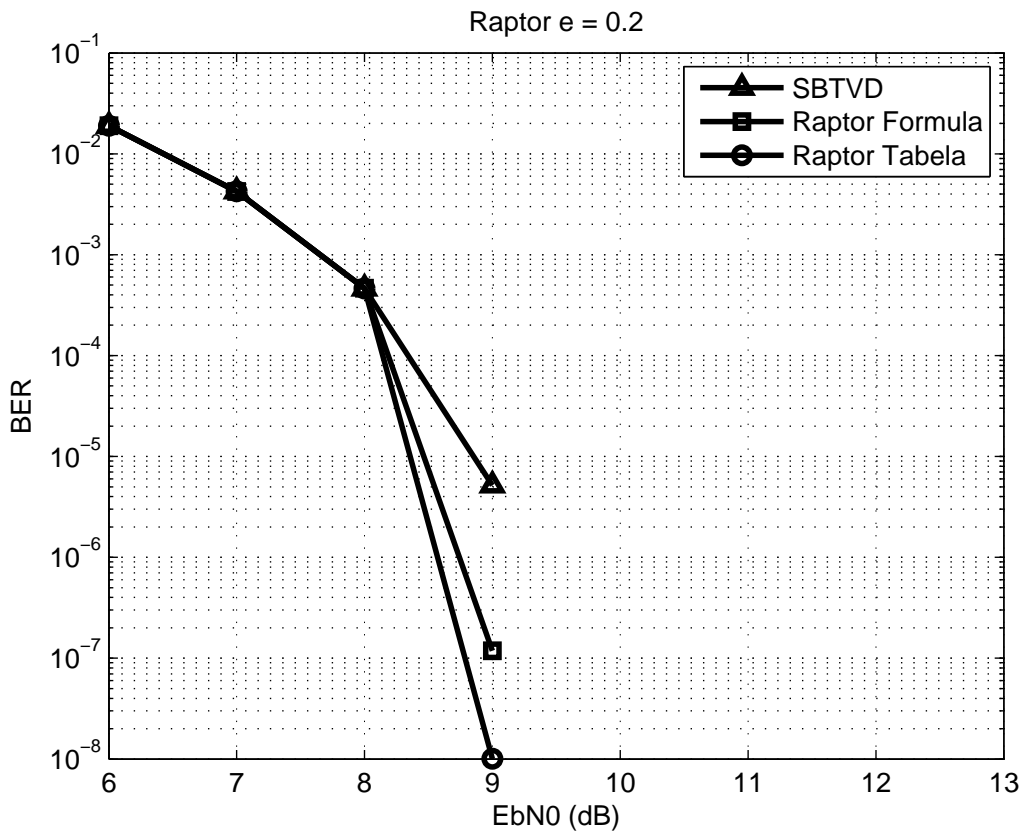


Figura 4.2: Comparação das distribuições de graus do LT para o Raptor aplicado ao SBTVD móvel, utilizando-se a Equação 3.19, e a tabela apresentada por de Shokrollahi (2006)

Segundo os poucos testes com distribuição de grau para LT que foram realizados durante o período desta pesquisa, os graus de distribuição dependem do canal. Como

esta pesquisa não foca melhorar ou achar a melhor distribuição de graus para o código enfraquecido LT, então utiliza-se a Tabela 4.1 para  $k = 65536$  para gerar as matrizes  $\mathbf{G}_{\text{LT}}$ .

É importante observar que ao gerar a matriz geradora com a distribuição aqui apresentada, não leva-se em conta que todos os símbolos estejam cobertos.

### 4.2.3 Metodologia para Decodificação dos Códigos Fontanais

Neste trabalho, para a decodificação do Código LT é utilizada a metodologia descrita no Capítulo 3.2.3, apresentada por Luby [12].

Já, a metodologia utilizada para a decodificação do Código Raptor não é o processo inverso da codificação LDPC e a codificação LT fraca. A metodologia utilizada para esta decodificação é através da resolução de equações lineares.

Desta forma, com a matriz de decodificação  $\mathbf{A}_{[i_1, i_2, \dots, i_r]}$  é resolvida a Equação 3.9, utilizando o método de eliminação Gaussiana.

Para a detecção de erros no Código Raptor, foi implementada a seguinte lógica: se houver erro na decodificação (não foi possível resolver a equação linear dada pela Equação 3.9), não é possível saber qual pacote está certo ou errado, o que, faz com que quantidade de erro a se considerar seja muito grande. Por isso, nesse caso leva-se em conta para a contagem de taxa de erro de *bit*, os pacotes recebidos do SBTVD sem serem decodificados pelo decodificador do Raptor.

Em caso de que seja possível a resolução da equação linear no Código Raptor, são levados em conta os erros apresentados pela decodificação do Código Raptor.

Para a detecção de erros no código LT, foi implementada a seguinte consideração: se a decodificação LT falhar (Capítulo 3.2.3, primeiro passo do algoritmo da decodificação), então leva-se em conta para a contagem de taxa de erro de *bit* os pacotes errados recebidos do SBTVD, mesma analogia que a decodificação do Raptor.

### 4.2.4 Distribuição de Graus para o Código LT

Para distribuição de graus do código LT, foi utilizada a distribuição Sóliton Robusta, do Capítulo 3.2.4, com  $c = 0,03$  e  $\delta = 0,1$ .

### 4.3 Tempo de Proteção

A proteção do código depende da taxa e do período de proteção[23]. A taxa é a relação entre os dados originais e a quantidade total de dados transmitidos, isto inclui os dados originais e os dados de correção. O período de proteção é a duração da informação codificada.

Segundo Gonzalez et al. (2009) [23], o tempo de proteção tem impacto na latência, como também no tempo de troca de canal. A latência é o tempo que o dado leva desde a entrada no transmissor até a decodificação da mídia. O tempo de troca de canal é o intervalo que o usuário leva desde a troca de canal até que a imagem apareça na televisão.

Quanto maior é o tempo de proteção, maior é o tempo de troca de canal e maior é a memória necessária para armazenar os dados. Segundo Rahrer et al. (2006) [24] e Fuchs e Farber (2008)[25], valores menores a 500 milissegundos de tempo de troca de canal são imperceptíveis; já, se maiores a dois segundos, são irritantes. Assim, a norma ABNT NBR 15608-2 (2007) [11] estabelece que o tempo apropriado é de 600 ms e o máximo é de menos de 1 segundo.

Segundo Faria et al. (2006 [26]) apud FURHT e AHSON, (2008 [7]), a taxa de *bit* média do ES é de 500 kbps, devido ao tamanho da tela do sistema DVB-H. Como o tamanho da tela do sistema DVB-H é o mesmo que a do SBTVD, então considera-se neste trabalho a taxa de *bit* ES de 500 kbps.

Sendo que a taxa de *bit* ES é de aproximadamente 500 kbps, ou 2  $\mu$ s por *bit*, e considerando que o tamanho do TS é fixo de 184 *bytes* sem o cabeçalho, então tem-se 1472 *bits* por cada TS, ou 2,9 ms por cada TS.

O tempo de atraso no transmissor afeta o tempo de troca de canal. Por isso, o tempo máximo de troca de canal de 1 segundo, como visto na norma ABNT NBR 15608-2 (2007) [11]. Considerando somente o atraso visto na Tabela 2.1, aplica-se neste trabalho o atraso da Tabela 4.2, levando em conta que se usa o modo 3.

Tabela 4.2: Especificação de atraso a ser considerada nas simulações

Tamanho do entrelaçador (I)	Atraso (ms)	$k$ símbolos fonte
4	379	214

Com a Tabela 4.2, neste trabalho, será utilizado tanto no Código Raptor, quanto no Código LT, o tamanho de entrelaçador 4 que conterà 214 símbolos fonte, ou originais. É importante lembrar, que a soma, de 379 ms, devido ao tamanho do entrelaçador, mais o tempo que leva armazenar 214 símbolos fonte para a codificação, equivale a 1 segundo.

## 4.4 O Simulador

O simulador utilizado para o trabalho é a ferramenta apresentada por Schwartz [18]. Este simulador recria o sistema de transmissão e de codificação de canal, por isso, tanto no transmissor, como no receptor, o autor exclui alguns blocos para facilitar a simulação. Mesmo assim, como neste trabalho precisa-se do sistema de transmissão e de codificação de canal do SBTVD, considera-se que não apresenta impacto significativo na pesquisa a exclusão de blocos como o de codificação de fonte, ajuste de atraso entre outros.

Como é necessário que os dados aproximem-se o máximo possível ao modelo *one-seg*, ou móvel, e como no simulador cada camada hierárquica é discriminada, então possibilita-se a extração de dados da camada central, que é a correspondente para o sistema móvel. Além disto, o deslocamento do móvel será simulado com a opção de modelo de canal Brasil A oferecido pela ferramenta de simulação.

As possibilidades de parâmetros que a ferramenta possui são:

- Taxas dos códigos convolucionais ( $1/2$ ,  $2/3$ ,  $3/4$ ,  $5/6$  e  $7/8$ );
- Modulações digitais (QPSK, 16-QAM e 64-QAM);
- Modos de operação (1, 2 e 3);
- Intervalo de guarda ( $1/4$ ,  $1/8$ ,  $1/16$  e  $1/32$ );
- Distribuição de segmentos entre as camadas:

1. Configuração 1: 13 segmentos para a camada “A”, zero para “B” e zero para “C”;



2. Configuração 2: 1 segmento para a camada “A”, 12 para “B” e zero para “C”;
  3. Configuração 3: 1 segmento para a camada “A”, 3 para “B” e 9 para “C”;
- Modelos de canais utilizados: Canal AWGN, Canal Brasil A, B, C, D e E fixo e móvel.

Desta forma, os parâmetros a serem utilizados para este trabalho dependem dos utilizados durante o período de pesquisa pelas emissoras de televisão em Curitiba e por algumas normas especificadas pela ABNT.

Assim, foram medidos, no dia 06 de outubro de 2011, os parâmetros de transmissão das emissoras no Brasil com o instrumento da Rohd & Schwarz, modelo R&S ETL TV *analyzer*. A tabela 4.3 mostra os resultados obtidos. Vide também as medições no Anexo A.

Tabela 4.3: Parâmetros de transmissão das emissoras no Brasil

Canal	Características	Camada A	Camada B
BAND	Modulação	QPSK	64QAM
	Taxa do código	2/3	3/4
	Entrelaçamento temporal	4	2
	Intervalo de guarda	1/16	
	Modo de operação	3 8K-FFT	
GLOBO	Modulação	QPSK	64QAM
	Taxa do código	2/3	3/4
	Entrelaçamento temporal	2	2
	Intervalo de guarda	1/16	
	Modo de operação	3 8K-FFT	
MERCOSUL	Modulação	QPSK	64QAM
	Taxa do código	2/3	3/4
	Entrelaçamento temporal	4	2
	Intervalo de guarda	1/16	
	Modo de operação	3 8K-FFT	
REDE RECORD	Modulação	QPSK	64QAM
	Taxa do código	2/3	3/4
	Entrelaçamento temporal	4	2
	Intervalo de guarda	1/16	
	Modo de operação	3 8K-FFT	
REDE VIDA	Modulação	QPSK	16QAM
	Taxa do código	2/3	2/3
	Entrelaçamento temporal	4	2
	Intervalo de guarda	1/8	
	Modo de operação	3 8K-FFT	

Assim também, a norma ABNT NBR para receptores *one-seg*, ou móveis, apresenta os seguintes parâmetros para o sistema móvel:

- Demodulação Digital (Demapeador) (QPSK, 16-QAM ou 64 QAM)[27];
- Taxa do Código Convolutacional: 1/2 [27].

Em combinação dos dados medidos e a norma ABNT, conclui-se aqui que os parâmetros utilizados nesta pesquisa serão:

- Modo 3;
- Modulação QPSK;
- Taxa do Código Convolutacional 1/2 e 2/3;
- Intervalo de Guarda 1/16;
- Comprimento do *Interleaving* 4.

## 4.5 Modelos de Canais Implementados no Simulador

Os canais que utilizam a ferramenta são padronizados para simulação fixa e são os canais Brasil A, B, C, D e E [28]. Esses canais representam vários cenários de medição com 6 *taps* de atraso e atenuações diferentes, exceto para o canal Brasil E com 3 *taps*.

O canal Brasil A representa um cenário normal (casa térrea), com antena externa. Já o canal B representa o mesmo canal A, mas com antena interna, ou uma antena externa, com defeitos[18]. O canal Brasil C considera regiões montanhosas e sem linha de visada. Já, o canal D é semelhante ao C, mas com antena interna, ou uma antena externa, com defeitos[18]. O canal E é um cenário de frequência única, o qual recebe sinais de mesma amplitude defasados, ou recebe impulsos em tempos diferentes[18]. Em diversos trabalhos, como, por exemplo, [29] e [30], são apresentados os canais Brasil móveis A, B, C, D e E, nos quais é acrescentada Frequência Doppler em um dos *taps*.

A Tabela 4.4 [28] a seguir mostra as especificações de atraso e atenuação para os diferentes canais Brasil.

Tabela 4.4: Especificações de atraso e atenuação para os diferentes canais Brasil

Nome	Descrição	Percurso 1	Percurso 2	Percurso 3	Percurso 4	Percurso 5	Percurso 6
Canal Brasil A	Atraso (us)	0	0,15	2,22	3,05	5,86	5,93
	Atenuação (dB)	0	13,8	16,2	14,9	13,6	16,4
	Espalhamento Doppler (Hz)	0	100	0	0	0	0
Canal Brasil B	Atraso (us)	0	0,3	3,5	4,4	9,5	12,7
	Atenuação (dB)	0	12	4	7	15	22
	Espalhamento Doppler (Hz)	0	0	10	0	0	0
Canal Brasil C	Atraso (us)	0	0,09	0,42	1,5	2,32	2,79
	Atenuação (dB)	2,8	0	3,8	0,1	2,5	1,3
	Espalhamento Doppler (Hz)	0	0	0	0	0	0
Canal Brasil D	Atraso (us)	0,1	0,63	2,22	3,05	5,86	5,93
	Atenuação (dB)	0,1	3,8	2,6	1,3	0	2,8
	Espalhamento Doppler (Hz)	10	0	0	0	0	0
Canal Brasil E	Atraso (us)	0	1	2	-	-	-
	Atenuação (dB)	0	0	0	-	-	-
	Espalhamento Doppler (Hz)	0	0	10	-	-	-

Na ferramenta de simulação apresentada por Schwartz (2011)[18], o método utilizado para simular o espectro Doppler do canal Brasil móvel é o *Restricted Jakes* do Matlab, onde é especificado um deslocamento de frequência Doppler  $f_d$  em Hertz.

Para poder obter frequências de espalhamento Doppler, com mínimas variações, como especificado na Tabela 4.4, o autor Schwartz [18], da ferramenta de simulação do SBTVD, viu-se obrigado a utilizar o método *Restricted Jakes*.

O método *Restricted Jakes* é comumente aplicado à antenas direcionais e é utilizado quando o espectro de potência não cobre a largura de banda completa entre  $-f_d$  a  $f_d$ . Neste caso, é especificada a frequência mínima normalizada de deslocamento Doppler  $f_{min,norm}$  e a frequência máxima normalizada de deslocamento Doppler  $f_{max,norm}$ , onde analiticamente  $f_{min,norm} = f_{min}/f_d$ ;  $f_{max,norm} = f_{max}/f_d$  e  $0 \leq f_{min,norm} \leq |f_{norm}| \leq f_{max,norm} \leq 1$ . Neste trabalho, utiliza-se  $f_{min,norm} = 0,95$  e  $f_{max,norm} = 1$  nos *taps* escolhidos com Frequência Doppler e, nos demais *taps*, de  $f_{min,norm} = 0$  e  $f_{max,norm} = 0,05$ . Assim a frequência Doppler de 100 Hz do canal Brasil A da tabela 4.4 oscila entre 95 Hz e 100 Hz e o número “0” que esta como frequência Doppler fica entre 5 Hz e 0 Hz.

Neste trabalho, foi utilizado o Canal Brasil A que representa um ambiente de recepção externa. E como a frequência Doppler é dada pela equação:

$$f_d = \frac{v}{\lambda} = \frac{v \cdot f}{c} \quad (4.1)$$

Encontra-se que 100 Hz para uma frequência de portadora de 600 MHz equivale a 180 Km/h, por isso opta-se neste trabalho por uma frequência Doppler de 34 Hz que equivale a 61,2 Km/h e uma frequência Doppler de 10 Hz que equivale a 18 Km/h, que seria uma velocidade mais adequada ao ambiente urbano.

#### 4.5.1 Decodificador Externo de Reed Solomon

Na ferramenta de simulação apresentada por Schwartz (2011)[18], para o decodificador externo (ou codificador Reed-Solomon) implementado, pode-se optar pelo decodificador analítico ou pelo decodificador externo da ferramenta computacional numérica de simulação Matlab.

No método analítico de Reed-Solomon, a BER é obtida através da Equação 4.2 [31]:

$$P_E \approx \frac{1}{n} \sum_{j=t+1}^n j \binom{n}{j} p^j (1-p)^{n-j} \quad (4.2)$$

Onde  $P_E$  é a taxa de erro de símbolos na decodificação do Reed Solomon,  $n$  é a quantidade de símbolos da palavra, neste caso 255;  $t$  é a quantidade de símbolos que podem ser corrigidos e  $p$  é a taxa de erro de símbolos dada pelo código decodificador interno, ou decodificador de código convolucional.

Os gráficos de taxa de erro de *bit* apresentados pela ferramenta de simulação são gerados com o decodificador externo analítico da Equação 4.2 (Vide [18]). Isto ajuda a simplicidade do sistema.

Na Figura 4.3, pode-se observar o método analítico e o decodificador Reed-Solomon do Matlab. Foram geradas três curvas: Cod. Conv. (1/2) + RS Est. 2D *Cubic Spline*, Cod. Conv. (1/2) + RS do Matlab 2D *Cubic Spline* e Cod. Conv. (1/2) Est. 2D *Cubic Spline*. Em todas elas foi utilizado o código convolucional 1/2. Na segunda curva, é aplicado o Reed Solomon do Matlab, na primeira curva é aplicada a Equação 4.2 e na terceira não foi aplicado o Reed Solomon.

Foi detetada que a diferença entre aplicação do método analítico, com a implementação do Reed Solomon do Matlab, deve-se ao fato que no código Matlab da ferramenta de

simulação do SBTVD do Schwartz [18], não foi levado em conta o desembaralhamento para gerar a curva analítica. Sendo assim, as duas curvas são bem parecidas.

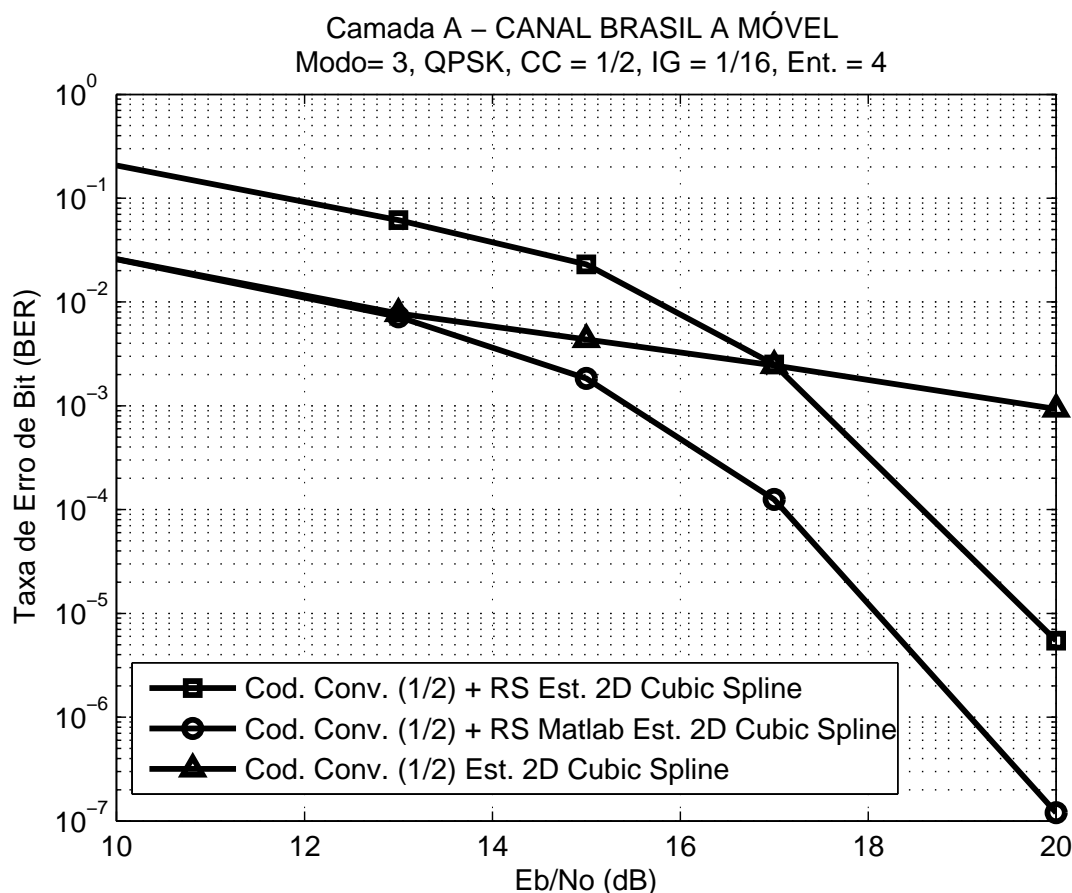


Figura 4.3: Taxa de Erro de *Bit* vs  $E_b/N_0$  para Taxa do Código Convolutacional 1/2, Comprimento do *Interleaving* 4, Canal Brasil A  $f_d = 34 H_z$

Para diminuir o tempo de processamento da ferramenta de simulação SBTVD, foi necessária a escolha de um modelo de perdas. Como nosso canal visto da camada de aplicação se comporta como um canal BEC, então é proposta, neste trabalho, a implementação do Modelo de Cadeia de Markov de Dois Estados.

## 4.6 Modelo de Cadeia de Markov de Dois Estados

O modelo da Cadeia de Markov de Dois Estados, também conhecido como modelo Gilbert-Elliot, foi escolhido devido a sua fácil implementação, baixa complexidade computacional e tendo em vista que o objetivo do trabalho não é encontrar o modelo que mais se adapta à resposta da ferramenta de simulação.

O modelo é estudado em vários trabalhos, como, por exemplo: [32], [33], [34], entre outros.

Considera-se o modelo de Cadeia de Markov de Dois Estados como foi introduzido por Gilbert[35], onde um dos estados é “B” para Bom e o outro estado é “R” para o Ruim, como mostra a Figura 4.4. Para a aplicação neste trabalho, um pacote ruim é interpretado como um erro de *bit*, ou de quadro TS, e um pacote bom é interpretado como um *bit* certo, ou um pacote TS sem erros. Dois parâmetros são definidos no modelo: “P” e “r”. “P” indica a probabilidade da transição do pacote bom para o pacote ruim e “r” a probabilidade de transição do pacote ruim para o pacote bom.

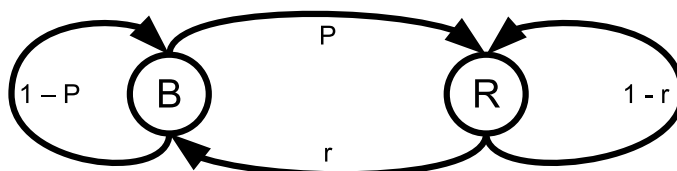


Figura 4.4: Modelo Gilbert-Elliot, ou modelo de Cadeia de Markov de Dois Estados

A matriz de transição é dada por:

$$A = \begin{pmatrix} 1 - P & P \\ r & 1 - r \end{pmatrix} \quad (4.3)$$

No simulador SBTVD, a sequência de *bits* originais é armazenada, como também a sequência de *bits* resultantes da simulação. Chama-se, neste trabalho, como sequência de erros a soma de módulo 2 dos *bits* originais com os *bits* resultantes da simulação.

Com a sequência de erros, para cada Taxa de Erro de *Bit* vs Energia de *Bit* por Taxa de Densidade Espectral de Potência do Ruído, (BER vs Eb/N0) (BER do inglês: *Bit Error Rate*) e Eb/N0 (Taxa de Energia de *Bit* por Densidade Espectral do Ruído), é utilizado o estimador do Modelo de Markov Escondido do Matlab, onde os erros são considerados como mudança de estado. Este estimador devolve a probabilidade de transição do modelo.

Com as probabilidades de transição, podem-se obter erros de *bit* e, também, erros de quadro TS, com a mesma taxa para cada Eb/N0 que a ferramenta de simulação propor-

ciona.

Para gerar as taxas de erros de *bit*, utiliza-se também o gerador de estados que provê o Matlab.

Na Figura 4.5, foram gerados uma única vez os erros com as probabilidades de transição obtidas do Simulador de Schwartz. Pode-se observar que a curva gerada pela metodologia de Gilbert-Elliot fica praticamente sobre a curva gerada pelo Simulador de Schwartz. Quanto mais vezes forem gerados os erros e retirada a média, a curva gerada pela metodologia de Gilbert-Elliot aproxima-se mais da gerada pelo Simulador de Schwartz. Este procedimento de gerar erros várias vezes acontece para obter-se um número mínimo de erros nas simulações apresentadas neste trabalho.

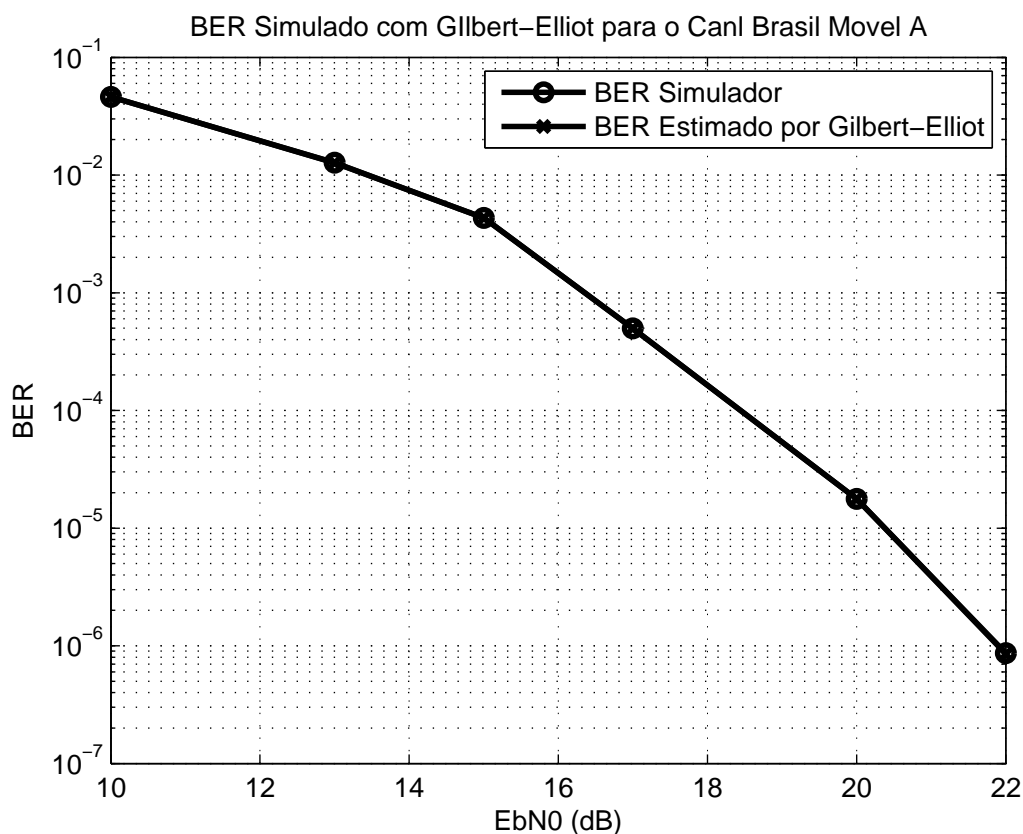


Figura 4.5: Modelamento dos resultados obtidos no Simulador de Schwartz com a ferramenta de Gilbert-Elliot

A curva da Figura 4.5 corresponde aos parâmetros: Camada Central, Modo 3, Modulação QPSK, Taxa de Código Convolutacional 2/3, Intervalo de Guarda 1/16, Comprimento de *Interleaving* 4, Canal Brasil A  $f_d = 34 H_z$ .

Para saber se o valor gerado pelo Gilbert-Elliot faz parte do valor simulado, acha-se o intervalo de confiança a partir da equação:

$$\mu \pm z \frac{\sigma_x}{\sqrt{n}} \quad (4.4)$$

Onde:

- $\mu$  é a media das amostras,
- $z$  é o número de desvios padrão da tabela normal,
- $\sigma_x$  desvio padrão,
- $n$  número de elementos da amostra.

A partir da equação 4.4 e com 50 amostras, obteve-se o intervalo de confiança que mostra as figuras 4.6, 4.7, 4.8, 4.9, 4.10 e 4.11.

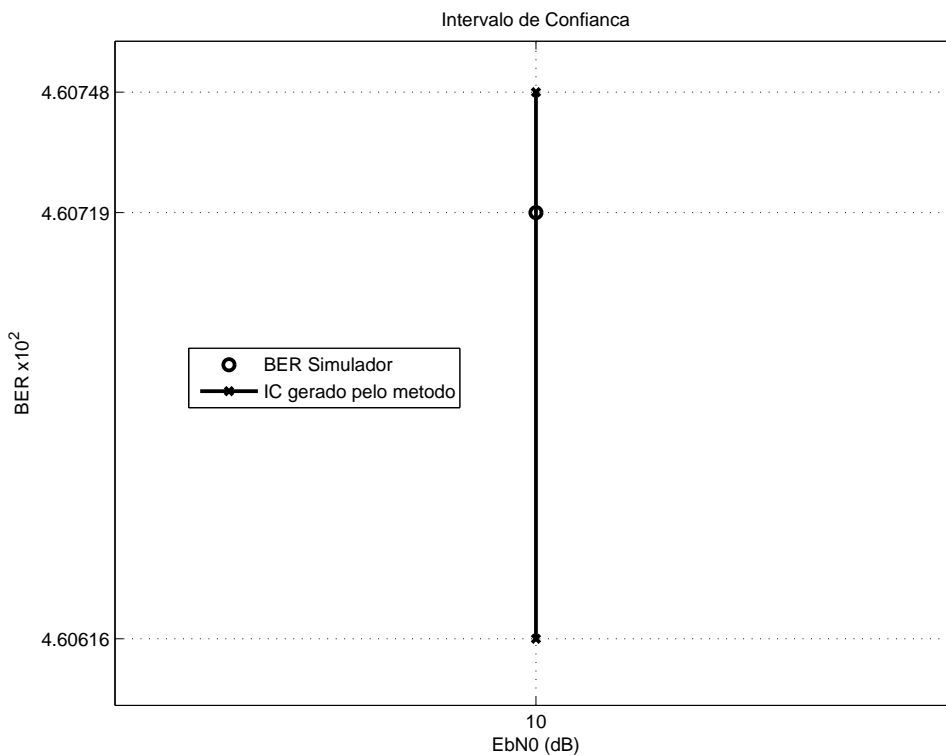


Figura 4.6: Intervalo de Confiança para  $E_b/N_0 = 10$



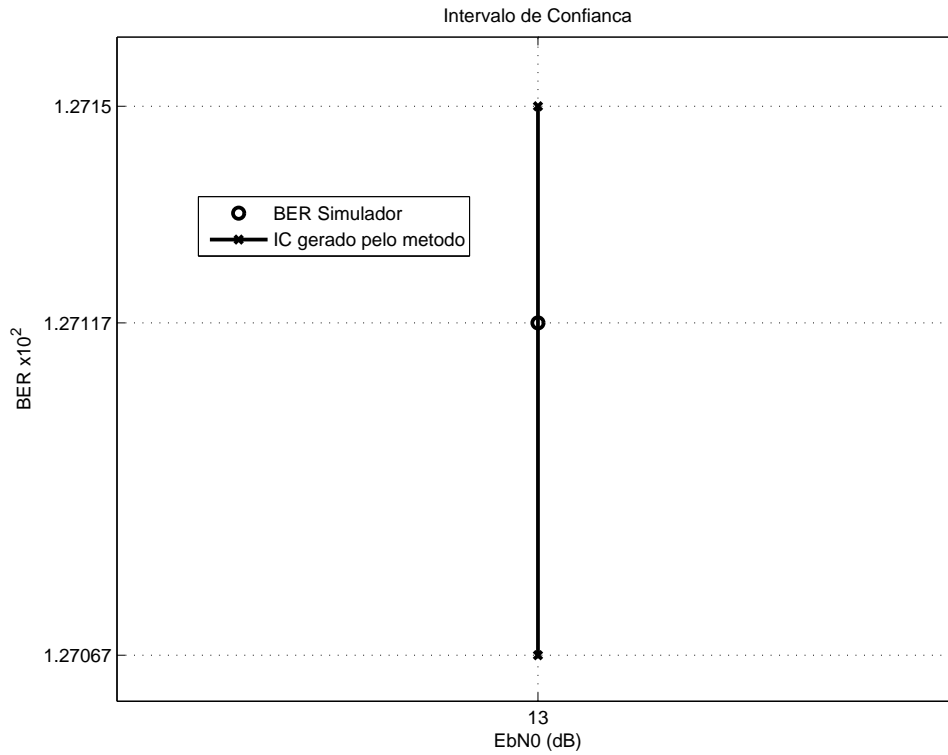


Figura 4.7: Intervalo de Confiança para  $EbN0 = 13$

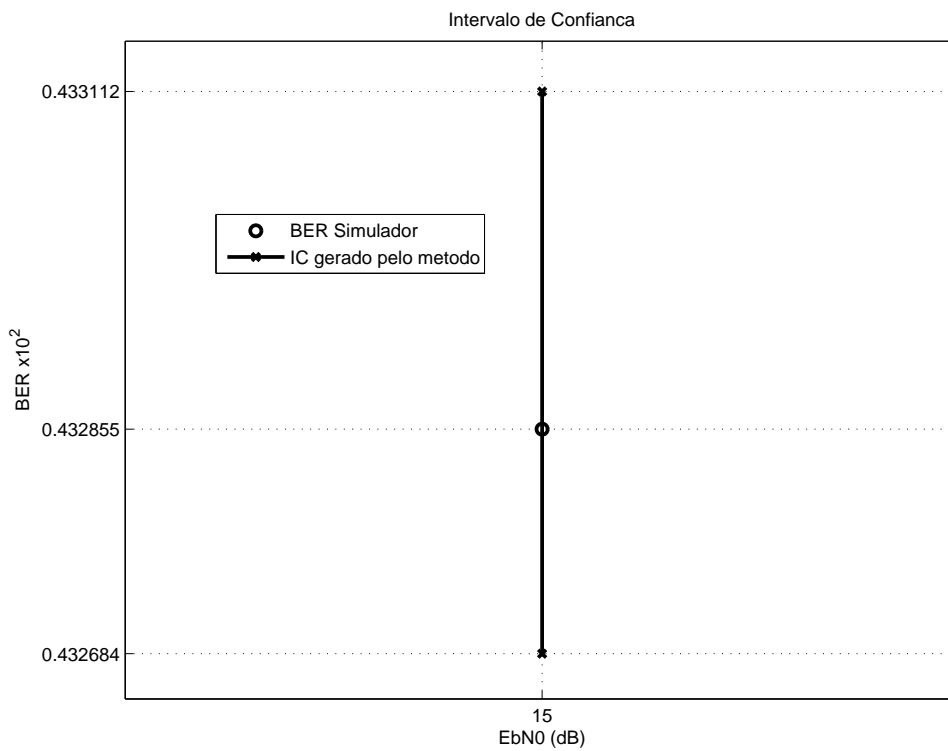


Figura 4.8: Intervalo de Confiança para  $EbN0 = 15$

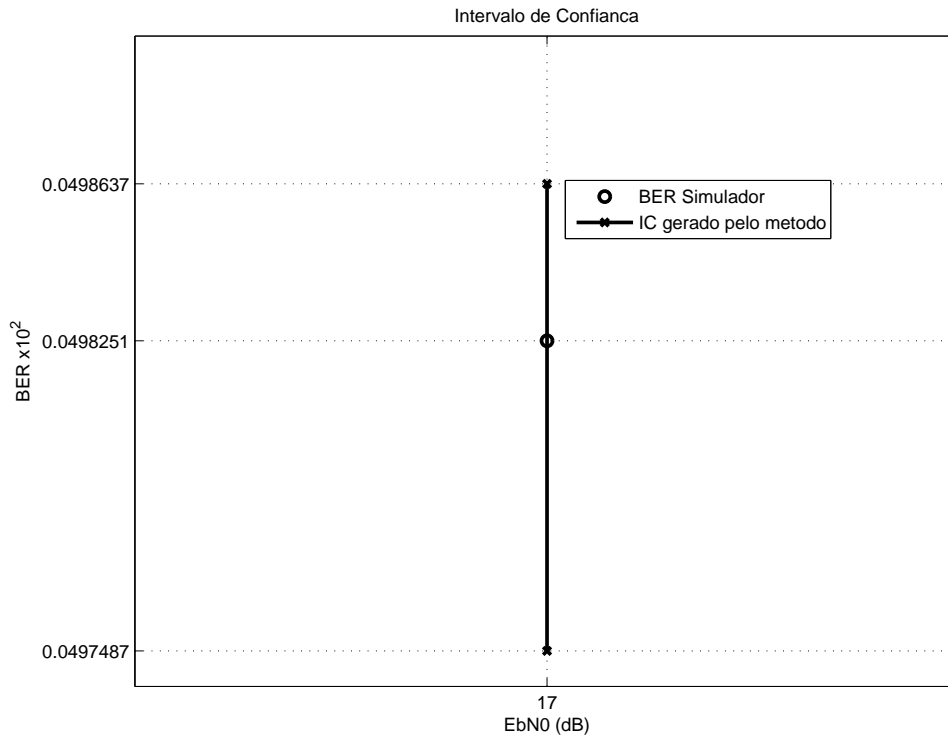


Figura 4.9: Intervalo de Confiança para  $E_b/N_0 = 17$

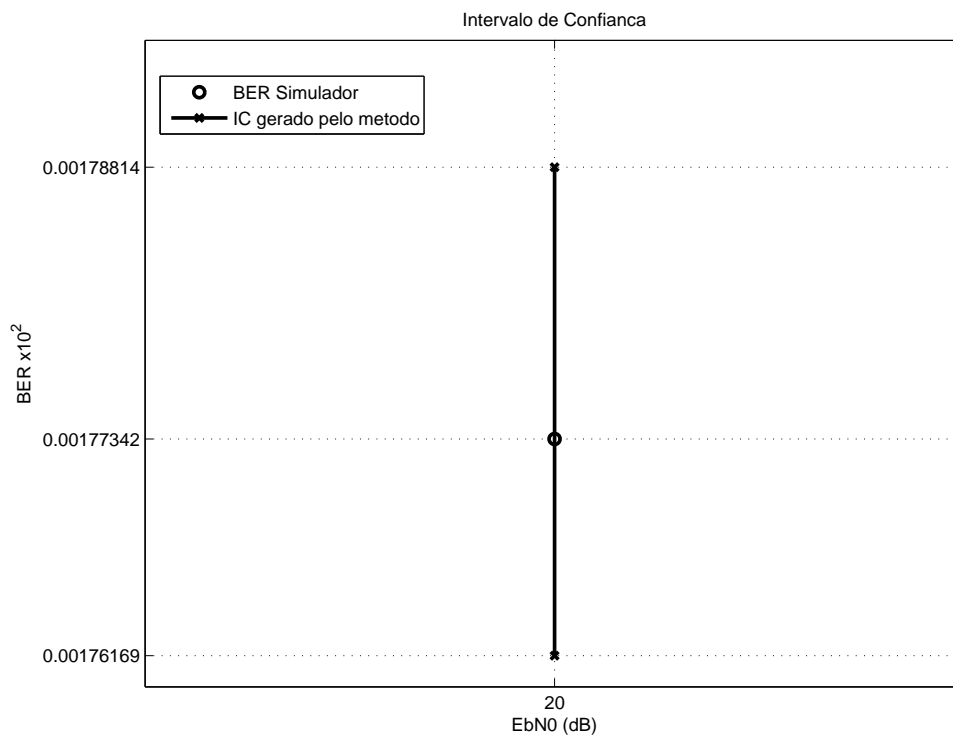


Figura 4.10: Intervalo de Confiança para  $E_b/N_0 = 20$

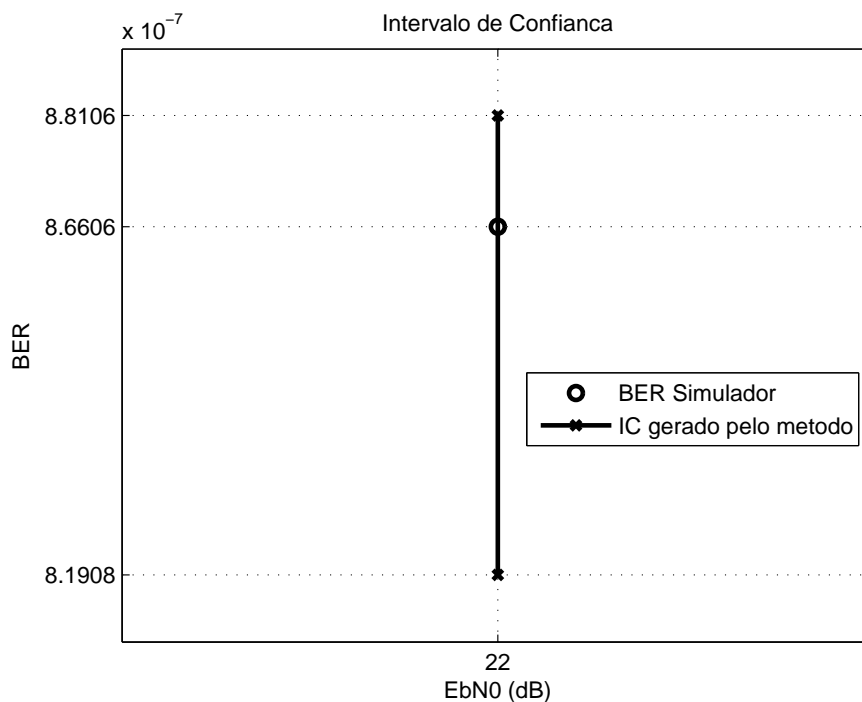


Figura 4.11: Intervalo de Confiança para  $E_b/N_0 = 22$

E com 95% de certeza, podemos dizer que os valores gerados pelo método Gilbert-Elliot são aproximados ao valor entregue pelo simulador. As aproximações equivalem a erros apresentados na seguinte tabela.

Tabela 4.5: Erro entre o ponto máximo e o ponto mínimo do intervalo de confiança e o valor entregue pelo Simulador

$E_b/N_0$	Erro+ ( $1.0e - 005$ )	Erro- ( $1.0e - 004$ )
10	0.289964208595450	0.102678070936224
13	0.325944670879694	0.050043560928872
15	0.256658271576005	0.017069190515733
17	0.038629241462388	0.007630947770413
20	0.014716008818674	0.001173143565101
22	0.001500087354216	0.000469784431957

Onde na tabela 4.5:

- Erro+ é a diferença entre o ponto máximo do Intervalo de Confiança e o valor entregue pelo Simulador;
- Erro- é a diferença entre o valor entregue pelo Simulador e o ponto mínimo do Intervalo de Confiança.

Desta forma, considera-se mínimo o erro apresentado pelo gerador Gilbert-Elliot e, assim, utiliza-se esta metodologia.

## **4.7 Considerações Finais**

Dadas todas as características relevantes para a arquitetura proposta, apresentar-se-á no seguinte capítulo os resultados desta pesquisa.

## CAPÍTULO 5

### RESULTADOS

#### 5.1 Introdução

Neste capítulo, serão apresentados os resultados obtidos na comparação entre os Código Raptor e o Código LT, assim como o desempenho do Código Raptor no SBTVD móvel para canal Brasil A, com Doppler de 10 Hz e 34 Hz, para diferentes taxas de códigos convolucionais.

#### 5.2 Comparação entre Código Raptor e Código LT

Os parâmetros do simulador SBTVD móvel para esta seção são: Camada Central, Modo 3, Modulação QPSK, Taxa do Código Convolucional 1/2, Intervalo de Guarda 1/16, Comprimento do *Interleaving* 4, Canal Brasil A  $f_d = 10$  Hz método *Restricted Jakes*,  $f_{min,norm} = 0,979$ ,  $f_{max,norm} = 1$  no 2 *tap* e nos demais *taps* de  $f_{min,norm} = 0$  e  $f_{max,norm} = 0,021$ . O desempenho de erro do sistema foi estimado utilizando-se como critério de parada a contagem de pelo menos 20 erros de *bit*.

Na Figura 5.1, compara-se o desempenho dos Códigos Fontanais, Raptor e LT, para um *overhead* de 20% do tamanho dos símbolos fonte, com  $c = 0,03$  e  $\delta = 0,1$ , na Distribuição Sóliton Robusta. Observa-se que o Raptor tem melhor desempenho, com BER de  $2,6249 \times 10^{-9}$  para o código Raptor e BER de  $2,7278 \times 10^{-7}$  para a codificação LT em 9 dB de  $E_b/N_0$ .

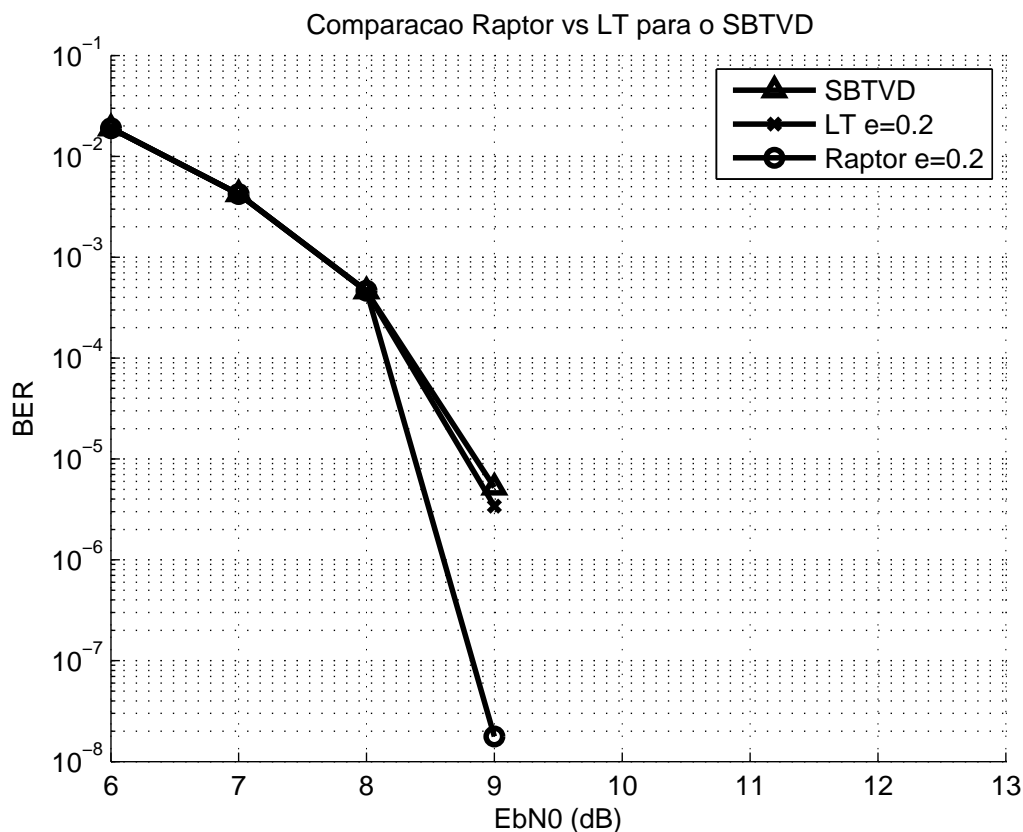


Figura 5.1: Desempenho dos Códigos Fontanaís Raptor e LT, para um *overhead* de 20% do tamanho dos símbolos fonte, com  $c = 0,03$  e  $\delta = 0,1$  na distribuição Sóliton Robusta

Com base nestas análises, é possível dizer que o Código Raptor teve melhor desempenho no cenário do SBTVD.

### 5.3 Desempenho do Código Raptor no SBTVD Móvel

Os parâmetros comuns do simulador SBTVD móvel para esta seção são: Camada Central, Modo 3, Modulação QPSK, Intervalo de Guarda 1/16, Canal Brasil A.

Na figura 5.2, segundo a norma ABNT [27] de Taxa do Código Convolutional 1/2, pode-se observar o desempenho do Código Raptor para um *overhead* de 20% do tamanho dos símbolos fonte. A curva corresponde à Comprimento do *Interleaving* 4 e Canal Brasil A  $f_d = 10$  Hz. Observa-se que se começa a ter ganhos a partir de 8 dB, e chega-se a  $3,0568 \times 10^{-9}$  de BER para 9 dB de  $E_b/N_0$ ; ou um ganho aproximado de 0,6 dB, para um BER de  $5,1801 \times 10^{-6}$ .

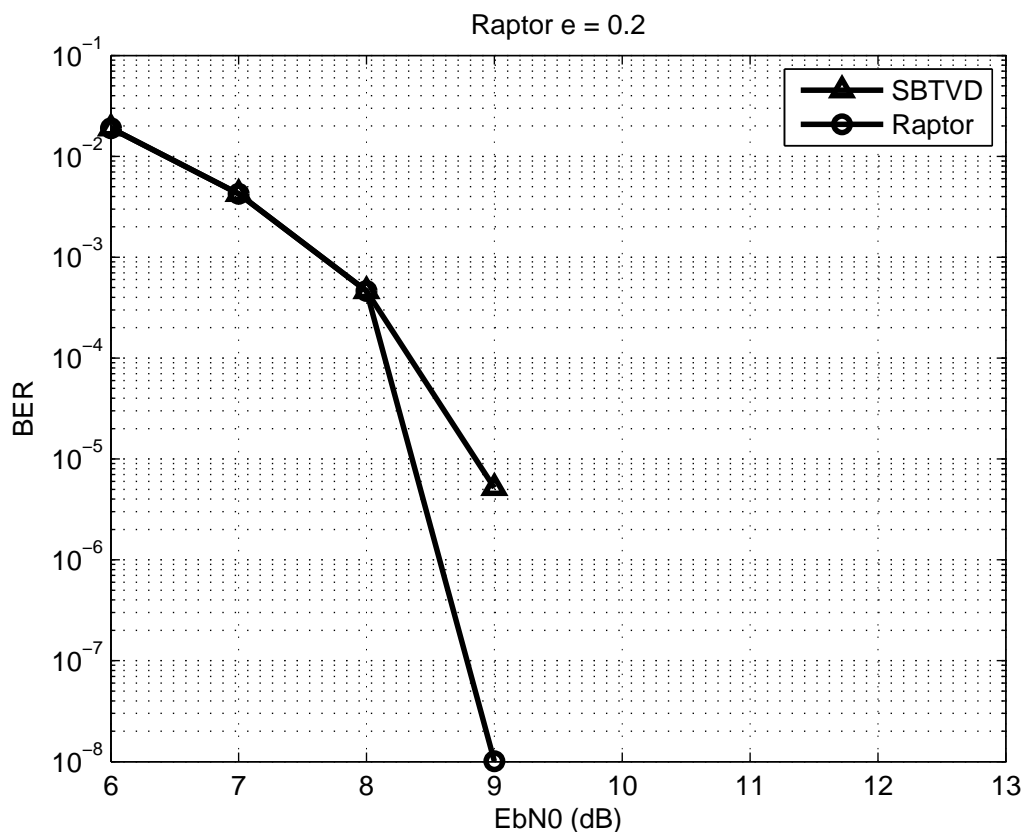


Figura 5.2: Desempenho do Código Raptor para um *overhead* de 20% do tamanho dos símbolos fonte. A curva corresponde à Taxa do Código Convolutacional 1/2, Comprimento do *Interleaving* 4 e Canal Brasil A  $f_d= 10$  Hz

Na figura 5.3, segundo a norma ABNT [27] de Taxa do Código Convolutacional 1/2, pode-se observar o desempenho do Código Raptor para um *overhead* de 50% do tamanho dos símbolos fonte. A curva corresponde à Comprimento do *Interleaving* 4 e Canal Brasil A  $f_d= 34$  Hz. Se começa a ter ganhos a partir de 15 dB, e chega-se a  $2,9175 \times 10^{-11}$  de BER para 20 dB de Eb/N0; ou um ganho aproximado de 2,5 dB, para um BER de  $1,7810 \times 10^{-7}$ .

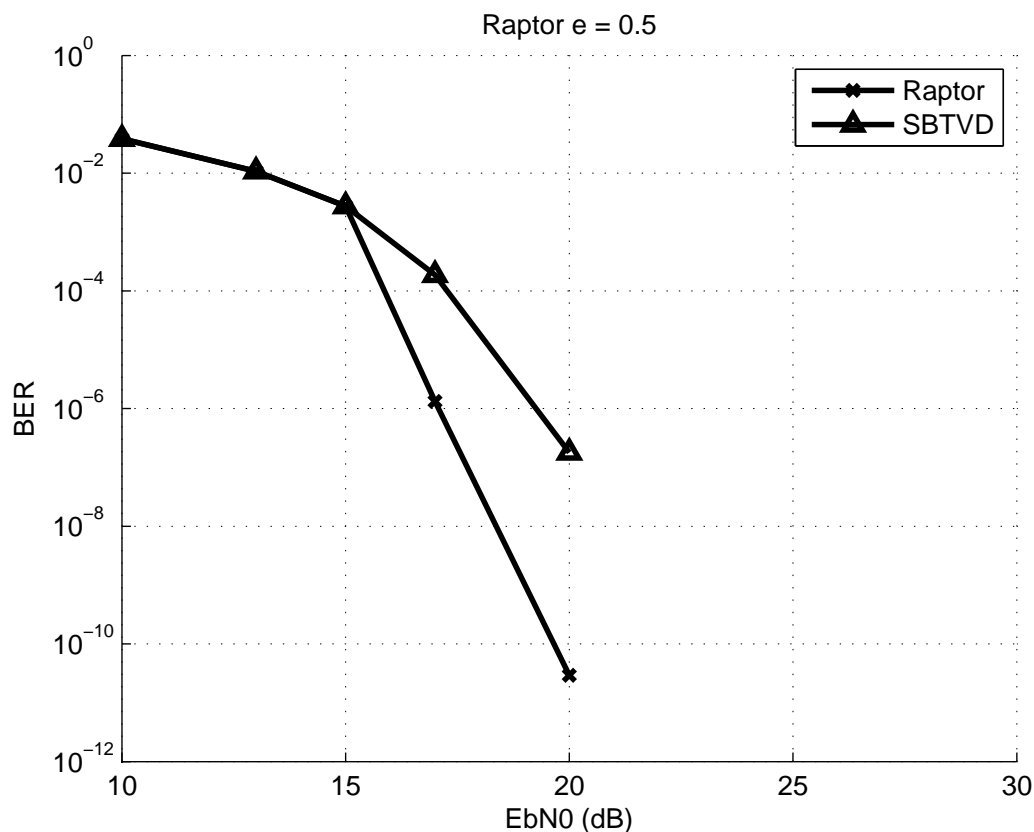


Figura 5.3: Desempenho do Código Raptor para um *overhead* de 50% do tamanho dos símbolos fonte. A curva corresponde à Taxa do Código Convolutacional 1/2, Comprimento do *Interleaving* 4 e Canal Brasil A  $f_d = 34$  Hz

Na Figura 5.4, segundo as medições da Tabela 4.4, realizadas em Curitiba, pode-se observar o desempenho do Código Raptor para um *overhead* de 50% do tamanho dos símbolos fonte. A curva corresponde à Taxa do Código Convolutacional 2/3, Comprimento do *Interleaving* 4, Canal Brasil A  $f_d = 34$  Hz. Observa-se que para 20 dB a taxa de erro de bit da recepção do SBTVD móvel é de  $1,2972 \times 10^{-6}$  e com a codificação AL-FEC tem-se  $3,3916 \times 10^{-9}$ . Com a codificação, tem-se um ganho aproximado de 3,2 dB, para um BER de  $1,2972 \times 10^{-6}$ .



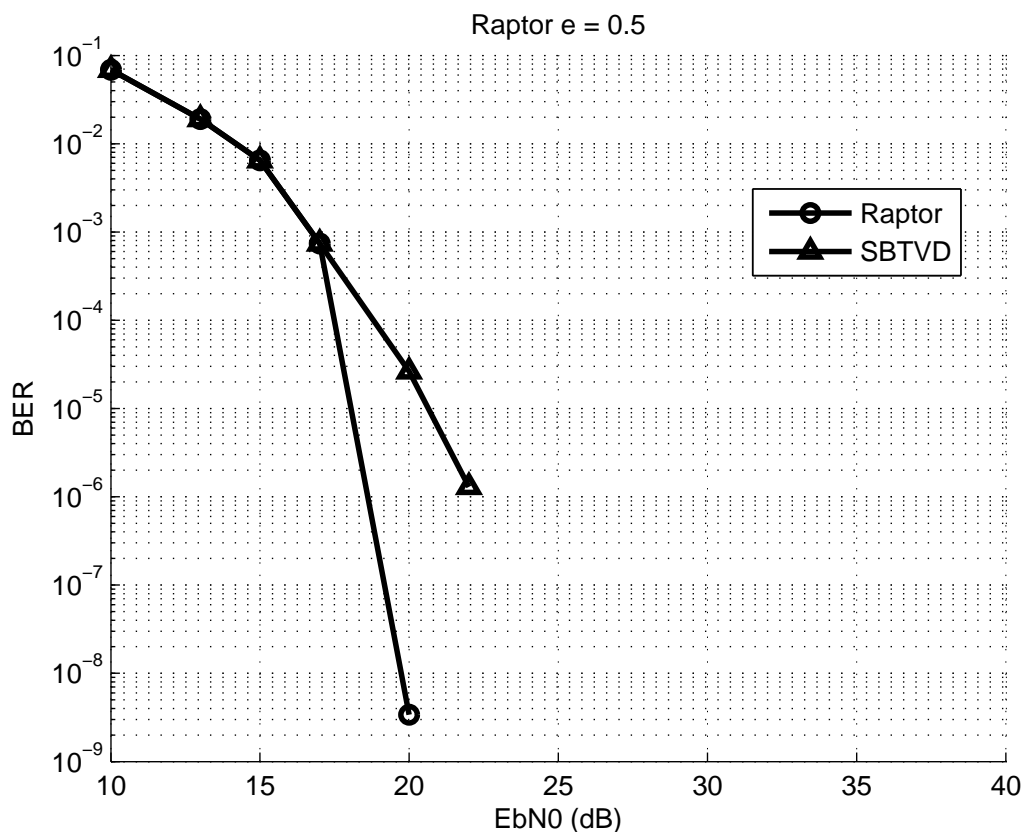


Figura 5.4: Desempenho do Código Raptor para um *overhead* de 50% do tamanho dos símbolos fonte. A curva corresponde à Taxa do Código Convolutivo 2/3, Comprimento do *Interleaving* 4 e Canal Brasil A  $f_d = 34$  Hz

## 5.4 Taxa de Decodificação com Sucesso

Nesta seção dos resultados, apresenta-se as taxas de decodificação com sucesso alcançadas para os seguintes parâmetros: Camada Central, Modo 3, Modulação QPSK, Intervalo de Guarda 1/16, Comprimento do *Interleaving* 4, Canal Brasil A.

No experimento, utiliza-se o *overhead* de Raptor de: 0,025, 0,05, 0,1, 0,2 e 0,5. Ou seja, se o *overhead* de 0,1 for utilizado, tem-se 10% a mais de símbolos do que o tamanho dos símbolos originais.

A metodologia utilizada é a seguinte: de 100 transmissões de 214 símbolos, subtrai-se de 1 da taxa de erros de decodificação. Lembrando que ocorre erro de decodificação quando não se pode resolver a Equação 3.9.

Na Tabela 5.1, apresentam-se os resultados para o caso em que temos Frequência

Doppler de 10 Hz. Quando utilizados os valores de 6, 7 e 8 dB, os mesmos apresentaram erros de decodificação em todas as transmissões; ou seja, precisava-se de mais de 0,5 de *overhead*. Para o valor de 9 dB, pode-se observar que 79% das transmissões não apresentaram erros na decodificação para 0,05 de *overhead* do Código Raptor, 31% para o *overhead* de 0,025, a maioria não apresenta erros para um *overhead* de 0,1 e não existe erro de decodificação para o *overhead* 0,2.

Tabela 5.1: Taxa máxima para Canal Brasil A, Frequência Doppler de 10 Hz

	<i>overhead</i> do Raptor			
EbN0 (dB)	0,025	0,05	0,1	0,2
9	31	79	99	100

Na Tabela 5.2, apresentam-se os resultados para o caso em que temos Frequência Doppler de 34 Hz. Quando utilizados os valores de 10, 13, 15 e 17 dB, os mesmos apresentaram erros de decodificação em todas as transmissões; ou seja, precisava-se de mais de 0,5 de *overhead*. Para o valor de 20 dB, pode-se observar que já não existe erro de decodificação para o *overhead* 0,3 ou 30%. Igualmente para o valor de 22 dB, que deixa de ter erros para um *overhead* de 0,3 ou 30%.

Tabela 5.2: Taxa máxima para Canal Brasil A, Frequência Doppler de 34 Hz, Código Convolutacional 1/2

	<i>overhead</i> do Raptor				
EbN0 (dB)	0,1	0,2	0,3	0,4	0,5
20	97,01	99,5	100	100	100
22	99	99,5	100	100	100

Na Tabela 5.3, apresentam-se os resultados para o caso em que temos Frequência Doppler de 34 Hz. Quando utilizados os valores de 10, 13, 15 e 17 dB, os mesmos apresentaram erros de decodificação em todas as transmissões; ou seja, precisava-se de mais de 0,5 de *overhead*. Para o valor de 20 dB, pode-se observar que já não existe erro

de decodificação para o *overhead* 0,2. Indistintamente para o valor de 22 dB, que deixa de ter erros para um *overhead* de 0,3.

Tabela 5.3: Taxa máxima para Canal Brasil A, Frequência Doppler de 34 Hz, Código Convolutacional 2/3

EbN0 (dB)	<i>overhead</i> do Raptor				
	0,1	0,2	0,3	0,4	0,5
20	95,52	100	100	100	100
22	97,51	99,5	100	100	100

## CAPÍTULO 6

### CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

Este trabalho teve como objetivo melhorar o desempenho na recepção móvel, através da utilização de um código na camada de aplicação do SBTVD.

Para as simulações, foram utilizados os seguintes parâmetros: Camada Central, Modo 3, Modulação QPSK, Taxa do Código Convolutacional 2/3 e 1/2 (segundo a norma da ABNT [27] e medições feitas em Curitiba, Brasil, tabela 4.3), Intervalo de Guarda 1/16, Comprimento do *Interleaving* 4, Canal Brasil A com Frequência Doppler de 10 Hz e 34 Hz, método *Restricted Jakes*.

Pode-se verificar no Capítulo 5, na comparação do Código Raptor com o Código LT, que o Código Raptor tem melhor desempenho comparado ao Código LT, com uma diferença de BER de  $2,6249 \times 10^{-9}$  para o Código Raptor e BER de  $2,7278 \times 10^{-7}$  para a codificação LT em 9 dB de  $E_b/N_0$ .

Foi realizada uma simulação para observar-se o desempenho do Código Raptor para *overhead* de 20%. Ou seja, sendo o número de símbolos originais 214, o tamanho de cada símbolo de 184 *bytes*, segundo foi visto no Capítulo 4.3, Frequência Doppler de 10 Hz e com taxa de código convolutacional de 1/2, segundo a norma da ABNT [27], observou-se que existe um ganho aproximado de 0,6 dB, para um BER de  $5,1801 \times 10^{-6}$ . Para a mesma simulação, observou-se que em 100 transmissões, não existe erro de decodificação.

Também foram realizadas simulações para observar-se o desempenho do Código Raptor para *overhead* de 50% para uma Frequência Doppler de 34Hz e com taxas do Código Convolutacional de 1/2 e 2/3, segundo a norma da ABNT [27] e medições feitas em Curitiba, Brasil, Tabela 4.3. Pode-se observar que para a curva que corresponde à Taxa do Código Convolutacional de 1/2, tem-se um ganho aproximado de 2,5 dB, para um BER de  $1,7810 \times 10^{-7}$  e BER de  $2,9175 \times 10^{-11}$  para 20 dB de  $E_b/N_0$ . Nesta configuração, não existem

erros de decodificação a partir de um *overhead* de 30%. A configuração correspondente à Taxa do Código Convolutacional de 2/3 tem um ganho aproximado de 3,2 dB, para um BER de  $1,2972 \times 10^{-6}$  e BER de  $3,3916 \times 10^{-9}$  para 20 dB de Eb/N0. Para esta última configuração, não existem erros de decodificação a partir de um *overhead* de 30%.

Este trabalho serve como pesquisa inicial de base para trabalhos futuros que envolvam pesquisas com AL-FEC aplicado ao SBTVD móvel. Uma sugestão de trabalho futuro é a implementação da arquitetura proposta neste trabalho, o Código Raptor em linguagem C ou C++ e finalmente a aplicação de uma plataforma real do SBTVD, com desvanecimento no canal. Outra sugestão bem interessante é a aplicação da comunicação cognitiva para melhorar a recepção móvel do SBTVD.

## BIBLIOGRAFIA

- [1] ILIEV, T. et al. Analysis and evaluation of reed-solomon codes in digital video broadcasting systems. In: *Wireless Telecommunications Symposium, 2008. WTS 2008*. [S.l.: s.n.], 2008. p. 92 –96. ISSN 1934-5070.
- [2] GOZALVEZ, D.; GOMEZ-BARQUERO, D.; STOCKHAMMER, T. Mobile reception of DVB-T services by means of AL-FEC protection. In: *Broadband Multimedia Systems and Broadcasting, 2009. BMSB '09. IEEE International Symposium on*. [S.l.: s.n.], 2009. p. 1 –5.
- [3] CUNCHE, M.; ROCA, V. *Improving the Decoding of LDPC Codes for the Packet Erasure Channel with a Hybrid Zyablov Iterative Decoding/Gaussian Elimination Scheme*. [S.l.], 2008. 19 p. Disponível em: <<http://hal.inria.fr/inria-00263682/en/>>.
- [4] GUTIERREZ, R. M.; SECO-GRANADOS, G. Efficiency comparison of LDPC-LDGM and Raptor codes for PL-FEC with very large block sizes. In: *Wireless Telecommunications Symposium, 2009. WTS 2009*. [S.l.: s.n.], 2009. p. 1 –6. ISSN 1934-5070.
- [5] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *ETSI DVB Bluebook A131: MPE-iFEC*. November 2008.
- [6] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *ETSI DVB Bluebook A054r4.1: Interaction channel for Satellite Distribution Systems*. January 2009.
- [7] FURHT, B.; AHSON, S. A. (Ed.). *Handbook of Mobile Broadcasting DVB-H, DMB, ISDB-T, AND MEDIAFLO*. [S.l.]: Auerbach Publications, 2008.
- [8] ABNT NBR 15602-3:2007. *Televisão digital terrestre - Codificação de vídeo, áudio e multiplexação Parte 3: Sistemas de multiplexação de sinais*. 2007.
- [9] ABNT NBR 15601:2007. *Televisão digital terrestre - Sistema de transmissão*. 2007.

- [10] AKAMINE, C. *Análise de desempenho do sistema de TV digital DVB-T com entrelaçador temporal em canais com ruído impulsivo*. Dissertação (Mestrado) — Universidade Estadual de Campinas, 2004.
- [11] ABNT NBR 15608-2. *Televisão digital terrestre - Guia de operação Parte 2: Codificação de vídeo, áudio e multiplexação - Guia para implementação da ABNT NBR 15602:2007*. 2010.
- [12] LUBY, M. LT Codes. In: *Proc. of the 43rd Annual IEEE Symp. on Foundation of Comp. Sc.* [S.l.: s.n.], 2002. p. 271–280.
- [13] SHOKROLLAHI, A. Raptor codes. *IEEE/ACM Trans. Netw.*, IEEE Press, Piscataway, NJ, USA, v. 14, p. 2551–2567, June 2006. ISSN 1063-6692. Disponível em: <<http://dx.doi.org/10.1109/TIT.2006.874390>>.
- [14] TORRES, C. M. C. *Transmissão de Imagem Através de Canal Ruidoso Usando Códigos LT*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2009.
- [15] MACKAY, D. J. C. *Information Theory, Inference & Learning Algorithms*. New York, NY, USA: Cambridge University Press, 2002. ISBN 0521642981.
- [16] PAIBA, F. S. *Códigos fontanaais para canais com apagamento*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, jul. 2009.
- [17] 3GPP TSG-SA WG4 S4-AHP238. *Specification Text for Systematic Raptor Forward Error Correction. PSM SWG, Sophia Antipolis*. France, Abril 2005.
- [18] SCHWARTZ, C. *Ferramenta de previsão de desempenho do sistema ISDB-T de TV digital*. Dissertação (Mestrado) — Universidade Federal do Paraná, 2011.
- [19] HU, X.-Y.; ELEFThERIOU, E.; ARNOLD, D. Regular and irregular progressive edge-growth tanner graphs. *Information Theory, IEEE Transactions on*, v. 51, n. 1, p. 386 –398, jan. 2005. ISSN 0018-9448.

- [20] PALANKI, R.; YEDIDIA, J. S. Rateless codes on noisy channels. In: *in Proc. Int. Symp. Inform. Theory*. [S.l.: s.n.], 2004. p. 37.
- [21] ETESAMI, O.; SHOKROLLAHI, A. Raptor Codes on Binary Memoryless Symmetric Channels. *IEEE Transactions on Information Theory*, v. 52, n. 5, p. 2033–2051, 2006.
- [22] SIVASUBRAMANIAN, B.; LEIB, H. Fixed-rate raptor code performance over correlated rayleigh fading channels. In: *Electrical and Computer Engineering, 2007. CCECE 2007. Canadian Conference on*. [S.l.: s.n.], 2007. p. 912 –915. ISSN 0840-7789.
- [23] GONZALEZ, D. et al. AL-FEC for Improved Mobile Reception of MPEG-2 DVB-T Transport Streams. *International Journal of Digital Multimedia Broadcasting*, v. 2009, p. 10, 2009. Article ID 614178.
- [24] RAHRER, T.; FIANDRA, R.; WRIGHT, S. *TR - 126 Triple-play Services Quality of Experience (QoE) Requirements*. [S.l.], December 2006.
- [25] FUCHS, H.; FARBER, N. Optimizing channel change time in IPTV applications. In: *Broadband Multimedia Systems and Broadcasting, 2008 IEEE International Symposium on*. [S.l.: s.n.], 2008. p. 1 –8.
- [26] FARIA, G. et al. DVB-H: Digital Broadcast Services to Handheld Devices. *Proceedings of the IEEE*, v. 94, n. 1, p. 194 –209, jan. 2006. ISSN 0018-9219.
- [27] ABNT NBR 15604:2007. *Televisão digital terrestre - Receptores*. 2007.
- [28] ITU RADIOCOMMUNICATION STUDY GROUPS: DOCUMENT 6E/TEMP/131-E. *Guidelines and Techniques for the Evaluation of DTTB Systems*. March 2003.
- [29] ISHINI, A. K.; AKAMINE, C. Técnicas de Estimacão de Canal para o Sistema ISDB-TB. *Revista de Radiodifuso*, v. 3 N.3, p. 151–157, 2009.
- [30] D’AGOSTINI, F. *Equalizao concorrente de canais no domnio frequncia para sistemas multiportadoras*. Dissertao (Mestrado) — Pontifcia Universidade Catlica do Rio Grande do Sul, 2007.



- [31] SKLAR, B. *Digital Communications: Fundamentals and Applications (2nd Edition)*. 2. ed. Prentice Hall, 2001. Hardcover. ISBN 0130847887. Disponível em: <<http://www.worldcat.org/isbn/0130847887>>.
- [32] WANG, H. S.; MOAYERI, N. Finite-state markov channel-a useful model for radio communication channels. vehicular technology. *IEEE Transactions on Vehicular Technology*, v. 44, p. 163–171, 1995.
- [33] WILHELMSSON, L. R. Evaluating the Performance of Raptor Codes for DVB-H by using the Gilbert-Elliott Channel. *VTC*, p. 1932–1936, 2007.
- [34] HASSLINGER, G.; HOHLFELD, O. The gilbert-elliott model for packet loss in real time services on the internet. 2008.
- [35] GILBERT, E. N. Capacity of a burst-noise channel. *Bell Syst. Tech. J.*, v. 39, p. 1253–1265, 1960.

## ANEXO A

No dia 06 de Outubro de 2011, mediram-se os parâmetros de transmissão das emissoras no Brasil com o instrumento da Rohd & Schwarz, modelo R&S ETL TV *analyzer*. A seguir mostra-se as impressões do monitor:

**R&S ETL Digital Overview** **S/N 102150, FW 2.12**

Ch: 22 UHF RF 521.142857 MHz ISDB-T 6 MHz

\* Att 0 dB

ExpLvl -37.50 dBm

Transmission and Multiplexing Configuration Control			
System Identification		ISDB-T	
Parameter Switching Indicator		Normal Value	
Emergency Alarm Broadcasting		Off	
Partial Reception		On	
Phase shift corr.		0x7	
Reserved Bits		0xffff	
TMCC	Layer A	Layer B	Layer C
Modulation	QPSK	64QAM	n/a
Code Rate	2/3	3/4	n/a
Time Interleaving	4	2	n/a
Nr. of Segments	1	12	-----
Lvl -64.7dBm   BER 8.6e-7   MER 26.0dB DEMOD MPEG			
Date: 6.OCT.2011 15:36:47			

Figura 6.1: Medições canal 22 Mercosul

R&S ETL Digital Overview S/N 102150, FW 2.12

Ch: 22 UHF RF 521.142857 MHz ISDB-T 6 MHz

\* Att 0 dB

ExpLvl -37.50 dBm

Fail	Limit	<	Results	<	Limit	Unit
Level	-60.0	*	-64.6		10.0	dBm
Sideband			Normal			
ISDB-T Mode			Mode 3, 8K-FFT			
Guard Interval			1/16			
Carrier Freq Offset	-30000.0		-82.5		30000.0	Hz
Bit Rate Offset	-100.0		-0.2		100.0	ppm
MER (total,rms)	24.0		26.2		-----	dB

		Layer A	Layer B	Layer C		
MER (Layer, rms)	24.0	26.9	25.7	-----		dB
BER before Viterbi		0.0e-7	2.9e-4	-----	1.0e-2	
BER before RS		0.0e-7	9.8e-7	-----	2.0e-4	
BER after RS		0.0e-6	0.0e-7	-----	1.0e-10	
Packet Error Ratio		0.0e-4	0.0e-5	-----	1.0e-8	
Packet Errors		0	0	-----	1	/s
MPEG TS Bitrate		0.4406	17.8428	-----		MBit/s

PA Lvl -64.6dBm | BER 9.5e-7 | MER 26.2dB DEMOD MPEG

Date: 6.OCT.2011 15:37:21

Figura 6.2: Medições canal 22 Mercosul

**R&S ETL Digital Overview** **S/N 102150, FW 2.12**

Ch: 34 UHF RF 593.142857 MHz ISDB-T 6 MHz

\* Att 0 dB  
ExpLvl -37.50 dBm

Transmission and Multiplexing Configuration Control			
System Identification		ISDB-T	
Parameter Switching Indicator		Normal Value	
Emergency Alarm Broadcasting		Off	
Partial Reception		On	
Phase shift corr.		0x7	
Reserved Bits		0xff	
TMCC	Layer A	Layer B	Layer C
Modulation	QPSK	64QAM	n/a
Code Rate	2/3	3/4	n/a
Time Interleaving	4	2	n/a
Nr. of Segments	1	12	-----
Lvl -65.6dBm   BER 4.3e-4   MER 22.2dB		DEM0D	MPEG
Date: 6.OCT.2011 15:32:10			

Figura 6.3: Medições canal 34 (RIC) Rede Record

R&S ETL Digital Overview S/N 102150, FW 2.12

Ch: 34 UHF RF 593.142857 MHz ISDB-T 6 MHz

\* Att 0 dB

ExpLvl -37.50 dBm

Fail	Limit	<	Results	<	Limit	Unit
Level	-60.0	*	-65.7		10.0	dBm
Sideband			Normal			
ISDB-T Mode			Mode 3, 8K-FFT			
Guard Interval			1/16			
Carrier Freq Offset	-30000.0		-5.0		30000.0	Hz
Bit Rate Offset	-100.0		0.0		100.0	ppm
MER (total,rms)	24.0	*	21.6		-----	dB

	Layer A	Layer B	Layer C		Unit
MER (Layer, rms)	24.0 *	22.5 *	21.8	-----	dB
BER before Viterbi	0.0e-7	* 1.6e-2		-----	1.0e-2
BER before RS	0.0e-7	* 6.4e-4		-----	2.0e-4
BER after RS	0.0e-6	0.0e-7		-----	1.0e-10
Packet Error Ratio	0.0e-4	0.0e-5		-----	1.0e-8
Packet Errors	0	0		-----	1 /s
MPEG TS Bitrate	0.4406	17.8428		-----	MBit/s

PA Lvl -65.7dBm | BER 6.2e-4 | MER 21.6dB DEMOD MPEG

Date: 6.OCT.2011 15:32:51

Figura 6.4: Medições canal 34 (RIC) Rede Record

**R&S ETL Digital Overview** **S/N 102150, FW 2.12**

Ch: 25 UHF RF 539.142857 MHz ISDB-T 6 MHz

\* Att 0 dB  
ExpLvl -37.50 dBm

Transmission and Multiplexing Configuration Control			
System Identification		ISDB-T	
Parameter Switching Indicator		Normal Value	
Emergency Alarm Broadcasting		Off	
Partial Reception		On	
Phase shift corr.		0x7	
Reserved Bits		0xffff	
TMCC	Layer A	Layer B	Layer C
Modulation	QPSK	16QAM	n/a
Code Rate	2/3	2/3	n/a
Time Interleaving	4	2	n/a
Nr. of Segments	1	12	-----
Lvl -75.6dBm   BER 8.7e-4   MER 12.3dB		DEMODO	MPEG
Date: 6.OCT.2011 15:30:25			

Figura 6.5: Medições canal 25 Rede Vida

R&S ETL Digital Overview S/N 102150, FW 2.12

Ch: 25 UHF RF 539.142857 MHz ISDB-T 6 MHz

\* Att 0 dB

ExpLvl -37.50 dBm

Fail	Limit	<	Results	<	Limit	Unit
Level	-60.0	*	-75.2		10.0	dBm
Sideband			Normal			
ISDB-T Mode			Mode 3, 8K-FFT			
Guard Interval			1/8			
Carrier Freq Offset	-30000.0		-4399.0		30000.0	Hz
Bit Rate Offset	-100.0		-14.4		100.0	ppm
MER (total,rms)	24.0	*	13.0		-----	dB

		Layer A	Layer B	Layer C		Unit
MER (Layer, rms)	24.0	* 19.0	* 12.7	-----		dB
BER before Viterbi		0.0e-7	* 4.3e-2	-----	1.0e-2	
BER before RS		0.0e-7	1.9e-4	-----	2.0e-4	
BER after RS		0.0e-6	0.0e-7	-----	1.0e-10	
Packet Error Ratio		0.0e-4	0.0e-5	-----	1.0e-8	
Packet Errors		0	0	-----	1	/s
MPEG TS Bitrate		0.4161	9.9860	-----		MBit/s

PA Lvl -75.2dBm | BER 1.8e-4 | MER 13.0dB DEMOD MPEG

Date: 6.OCT.2011 15:29:58

Figura 6.6: Medições canal 25 Rede Vida

R&S ETL Digital Overview S/N 102150, FW 2.12

Ch: 38 UHF RF 617.142857 MHz ISDB-T 6 MHz

\* Att 0 dB  
ExpLvl -37.50 dBm

Transmission and Multiplexing Configuration Control			
System Identification	ISDB-T		
Parameter Switching Indicator	Normal Value		
Emergency Alarm Broadcasting	Off		
Partial Reception	On		
Phase shift corr.	0x7		
Reserved Bits	0xff		
TMCC	Layer A	Layer B	Layer C
Modulation	QPSK	64QAM	n/a
Code Rate	2/3	3/4	n/a
Time Interleaving	4	2	n/a
Nr. of Segments	1	12	-----
Lvl -70.9dBm   BER 8.2e-3   MER 18.2dB		DEMOD	MPEG
Date: 6.OCT.2011 15:35:07			

Figura 6.7: Medições canal 38 Bandeirantes



R&S ETL Digital Overview S/N 102150, FW 2.12

Ch: 38 UHF RF 617.142857 MHz ISDB-T 6 MHz

\* Att 0 dB

ExpLvl -37.50 dBm

Fail	Limit	<	Results	<	Limit	Unit
Level	-60.0	*	-70.9		10.0	dBm
Sideband			Normal			
ISDB-T Mode			Mode 3, 8K-FFT			
Guard Interval			1/16			
Carrier Freq Offset	-30000.0		-5.6		30000.0	Hz
Bit Rate Offset	-100.0		-0.6		100.0	ppm
MER (total,rms)	24.0	*	18.0		-----	dB

		Layer A	Layer B	Layer C		Unit
MER (Layer, rms)	24.0	* 23.1	* 18.1	-----		dB
BER before Viterbi		0.0e-7	* 5.1e-2	-----	1.0e-2	
BER before RS		0.0e-7	* 9.8e-3	-----	2.0e-4	
BER after RS		0.0e-6	* 6.6e-4	-----	1.0e-10	
Packet Error Ratio		0.0e-4	* 1.1e-1	-----	1.0e-8	
Packet Errors		0	* 584	-----	1	/s
MPEG TS Bitrate		0.4406	17.8428	-----		MBit/s

PA Lvl -70.9dBm | BER 9.5e-3 | MER 18.0dB DEMOD MPEG

Date: 6.OCT.2011 15:34:37

Figura 6.8: Medições canal 38 Bandeirantes

R&S ETL Digital Overview S/N 102150, FW 2.12

Ch: 41 UHF RF 635.142857 MHz ISDB-T 6 MHz

\* Att 0 dB

ExpLvl -37.50 dBm

Transmission and Multiplexing Configuration Control			
System Identification	ISDB-T		
Parameter Switching Indicator	Normal Value		
Emergency Alarm Broadcasting	Off		
Partial Reception	On		
Phase shift corr.	0x7		
Reserved Bits	0xff		
TMCC	Layer A	Layer B	Layer C
Modulation	QPSK	64QAM	n/a
Code Rate	2/3	3/4	n/a
Time Interleaving	2	2	n/a
Nr. of Segments	1	12	-----
Lvl -62.8dBm   BER 6.1e-8   MER 23.7dB		DEM0D	MPEG
Date: 6.OCT.2011 15:22:29			

Figura 6.9: Medições canal 41 Rede Paranaense de Comunicação TV

R&S ETL Digital Overview S/N 102150, FW 2.12

Ch: 41 UHF RF 635.142857 MHz ISDB-T 6 MHz

\* Att 0 dB

ExpLvl -37.50 dBm

Fail	Limit	<	Results	<	Limit	Unit
Level	-60.0	*	-62.6		10.0	dBm
Sideband			Normal			
ISDB-T Mode			Mode 3, 8K-FFT			
Guard Interval			1/16			
Carrier Freq Offset	-30000.0		-6.9		30000.0	Hz
Bit Rate Offset	-100.0		0.0		100.0	ppm
MER (total,rms)	24.0		24.7		-----	dB

		Layer A	Layer B	Layer C		Unit
MER (Layer, rms)	24.0	* 19.8	25.4	-----		dB
BER before Viterbi		1.0e-7	5.5e-3	-----	1.0e-2	
BER before RS		0.0e-7	3.1e-8	-----	2.0e-4	
BER after RS		0.0e-6	0.0e-8	-----	1.0e-10	
Packet Error Ratio		0.0e-4	0.0e-6	-----	1.0e-8	
Packet Errors		0	0	-----	1	/s
MPEG TS Bitrate		0.4406	17.8428	-----		MBit/s

PA Lvl -62.6dBm | BER 3.0e-8 | MER 24.7dB DEMOD MPEG

Date: 6.OCT.2011 15:23:47

Figura 6.10: Medições canal 41 Rede Paranaense de Comunicação TV