

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**  
Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial

---

**DISSERTAÇÃO**  
apresentada à UTFPR  
para obtenção do grau de

**MESTRE EM CIÊNCIAS**

por

**TAMARA RODRIGUES ANDRICH**

---

**PROPOSTA DE ESQUEMAS ARQ HÍBRIDO USANDO CÓDIGOS TURBO PARA OS  
CANALIS SISO E MIMO**

---

Banca Examinadora:

Presidente e Orientador:

Prof. Dr. RICHARD DEMO SOUZA

UTFPR

Examinadores:

Prof. Dr. WALTER GODOY JÚNIOR

UTFPR

Prof. Dr. MARCELO EDUARDO PELLEZ

PUC-PR

Prof. Dr. EVELIO MARTÍN GARCÍA FERNÁNDEZ

UFPR

Curitiba, 20 de dezembro de 2006.

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

**TAMARA RODRIGUES ANDRICH**

**PROPOSTA DE ESQUEMAS ARQ HÍBRIDO USANDO CÓDIGOS TURBO PARA OS  
CANALIS SISO E MIMO**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do grau de "Mestre em Ciências"-  
Área de Concentração: Telemática

Orientador: Prof. Dr. Richard Demo Souza  
Co-Orientador: Prof. Dr. Walter Godoy Júnior

Curitiba  
2006

# *Agradecimentos*

Primeiramente gostaria de agradecer ao meu orientador Richard Demo Souza pelas sugestões, dedicação, paciência e amizade durante o desenvolvimento desta dissertação. Suas idéias, seu incentivo e sua criatividade ajudaram-me muito na elaboração deste trabalho.

Ao meu co-orientador Walter Godoy Júnior, o qual primeiro recebeu-me na instituição, oferecendo-me apoio e infraestrutura. Guardo bons sentimentos, amizade e gratidão.

Ao Prof. Marcelo Eduardo Pellenz, o qual contribui em uma parte do trabalho, mas principalmente por ter-me despertado a curiosidade e o interesse em uma área antes desconhecida.

Aos meus amigos e colegas do CITEC e do LASD, pelas brincadeiras e momentos de descontração.

Às agências financiadoras CNPq e CAPES, pelo auxílio material.

Aos meus pais Tereza e Tacílio e as minhas irmãs Taiza e Tânia, pelo amor e carinho. E em especial a Tânia pela companhia, o tempo que moramos juntas em Curitiba, será lembrado como momentos muito especiais.

E por último, mas não o menos importante ao meu marido Roberto Andrich por seu altruísmo e apoio incondicional durante estes dois anos de mestrado em que estive ausente do nosso lar. Pensar em você era um alívio para as minhas angústias.

# *Sumário*

<b>Lista de Figuras</b>	p. vii
<b>Lista de Tabelas</b>	p. x
<b>Lista de Abreviaturas</b>	p. xii
<b>1 Introdução</b>	p. 16
1.1 Estrutura da Dissertação . . . . .	p. 17
1.2 Contribuições . . . . .	p. 18
<b>2 Modelo do Sistema de Comunicação</b>	p. 20
2.1 Modelo do Sistema . . . . .	p. 20
2.2 Modelos Matemáticos para os Canais de Comunicação . . . . .	p. 21
2.2.1 Canal AWGN . . . . .	p. 22
2.2.2 Canal Rayleigh . . . . .	p. 22
2.3 Diversidade Espacial . . . . .	p. 23
2.3.1 Modelos Matemáticos dos Canais SISO, SIMO, MISO e MIMO . . . . .	p. 25
2.3.2 Efeito da Diversidade Espacial . . . . .	p. 27
2.4 Técnicas Fundamentais de Controle de Erros . . . . .	p. 28
2.4.1 FEC . . . . .	p. 28
2.4.2 ARQ . . . . .	p. 31
2.4.3 ARQ Híbrido . . . . .	p. 34
2.5 Comentários Finais . . . . .	p. 35

<b>3</b>	<b>Códigos Turbo</b>	p. 36
3.1	Introdução . . . . .	p. 36
3.2	Codificador Turbo . . . . .	p. 38
3.2.1	Codificadores Sistemáticos Recursivos . . . . .	p. 39
3.2.2	Interleaver . . . . .	p. 40
3.2.3	Puncionamento . . . . .	p. 42
3.3	Decodificador . . . . .	p. 42
3.3.1	Algoritmo BCJR Modificado . . . . .	p. 43
3.4	Desempenho do Código Turbo . . . . .	p. 48
3.4.1	Capacidade de Canal . . . . .	p. 48
3.4.2	Efeito do Número de Iterações no Desempenho . . . . .	p. 49
3.4.3	Efeito do Tamanho do Interleaver no Desempenho . . . . .	p. 50
3.4.4	Desempenho do Código Convolutacional Versus desempenho do Código Turbo . . . . .	p. 51
3.5	Comentários Finais . . . . .	p. 52
<b>4</b>	<b>Esquema de ARQ Híbrido Baseado nos Códigos Convolutacionais Recursivos e na Decodificação Turbo</b>	p. 53
4.1	Introdução . . . . .	p. 53
4.2	Modelo do Sistema . . . . .	p. 54
4.2.1	Segunda Transmissão . . . . .	p. 56
4.2.2	Terceira Transmissão . . . . .	p. 57
4.2.3	Quarta transmissão . . . . .	p. 57
4.2.4	Transmissões Subseqüentes . . . . .	p. 58
4.3	Resultados de Simulação . . . . .	p. 58
4.4	Análise da Complexidade Computacional . . . . .	p. 59
4.5	Comentários Finais . . . . .	p. 67

<b>5</b>	<b>Esquema de ARQ Híbrido Utilizando RSM e Códigos Turbo</b>	p. 68
5.1	Introdução . . . . .	p. 68
5.2	Modelo do Sistema . . . . .	p. 69
5.3	Esquema ARQ Híbrido Usando RSM . . . . .	p. 71
5.4	Throughput do TCP sobre Links Sem fio Usando RSM e Códigos Turbo . . .	p. 73
5.4.1	Avaliação do Throughput do TCP Usando o Esquema ARQ Híbrido .	p. 75
5.5	Comentários Finais . . . . .	p. 79
<b>6</b>	<b>Conclusão e Trabalhos Futuros</b>	p. 80
	<b>Referências Bibliográficas</b>	p. 82

# *Lista de Figuras*

2.1	Diagrama de Blocos do Modelo do Sistema de Comunicação. . . . .	p. 21
2.2	Canal AWGN. . . . .	p. 22
2.3	Canal Rayleigh. . . . .	p. 23
2.4	Sistemas SISO, MISO, SIMO e MIMO. . . . .	p. 25
2.5	Efeito do aumento do número de antenas na variação do nível da potência do sinal recebido. . . . .	p. 28
2.6	Código convolucional de taxa 1/2 e m=4. . . . .	p. 30
2.7	Diagrama de tráfego de pacotes do protocolo ARQ <i>stop-and-wait</i> . . . . .	p. 32
2.8	Diagrama de tráfego de pacotes do protocolo ARQ <i>go-back-N</i> . . . . .	p. 33
2.9	Diagrama de tráfego de pacotes do protocolo ARQ <i>Selective Repeat</i> . . . . .	p. 34
3.1	Diagrama de Blocos do Codificador Turbo. . . . .	p. 38
3.2	Codificador Convolucional Sistemático não Recursivo. . . . .	p. 40
3.3	Codificador Convolucional Sistemático Recursivo. . . . .	p. 41
3.4	Diagrama de Blocos do Decodificador Turbo. . . . .	p. 43
3.5	Cálculo de $\alpha$ e $\beta$ . . . . .	p. 45
3.6	Eficiência Espectral por $E_b/N_0$ para a Modulação BPSK. . . . .	p. 49
3.7	Taxa de erro de Bit no canal AWGN com diferentes números de iterações simulado com codificador turbo de taxa 1/3 e L=1024. . . . .	p. 50
3.8	Taxa de erro de Bit (BER) pela relação $E_b/N_0$ para o codificador turbo de taxa 1/3 e oito iterações. . . . .	p. 51
3.9	Taxa de erro de bit (BER) por $E_b/N_0$ para o código Turbo e o Código Convolucional de taxa 1/3. . . . .	p. 52
4.1	Diagrama de blocos de um sistema de comunicação empregando o protocolo ARQ. . . . .	p. 55



4.2	Diagrama de blocos com a estratégia de transmissão do esquema proposto, considerando a primeira e a segunda transmissões. RCC significa um codificador convolucional recursivo, enquanto que $\pi$ é um <i>interleaver</i> de tamanho $L$ . . . . .	p. 56
4.3	$FER$ por $E_b/N_0$ para a primeira e segunda transmissão considerando o esquema proposto e o método em [34]. . . . .	p. 59
4.4	$\Gamma = 1 - FER$ por $E_b/N_0$ depois da segunda transmissão para o esquema ARQ/FEC híbrido regular usando código turbo, o esquema ARQ/FEC híbrido modificado usando o código turbo em [34], e o método proposto neste capítulo. . . . .	p. 60
4.5	$\Gamma = 1 - FER$ depois da terceira transmissão para o esquema proposto. Duas estratégias são consideradas: i) a transmissão dos símbolos de paridade $x^{p1}$ ; ii) a transmissão dos símbolos sistemáticos $x^s$ . Na curva também é mostrado o desempenho depois da segunda transmissão. . . . .	p. 60
4.6	Módulo da treliça para o código convolucional (3,2,2). O número total de símbolos é 48, então a complexidade da treliça é 48/2 símbolos por bit . . . . .	p. 61
4.7	Relação de complexidade entre o esquema proposto e o método em [34], $C_{new}/C_{Nar}$ , como função do número de transmissões, para o caso de $n = 2$ , $k = 1$ , $n' = 3$ , $k' = 1$ , e $m = 3$ . . . . .	p. 62
4.8	Número de vezes que o esquema proposto e o método em [34] teve que utilizar a 1 <sup>a</sup> , 2 <sup>a</sup> , 3 <sup>a</sup> , 4 <sup>a</sup> e 5 <sup>a</sup> transmissão para enviar 10,000 pacotes. $E_b/N_0 = -1$ dB. . . . .	p. 63
4.9	Número de vezes que o esquema proposto e o método em [34] teve que utilizar a 1 <sup>a</sup> , 2 <sup>a</sup> , 3 <sup>a</sup> , 4 <sup>a</sup> e 5 <sup>a</sup> transmissão para enviar 10,000 pacotes. $E_b/N_0 = 0$ dB. . . . .	p. 64
4.10	Número de vezes que o esquema proposto e o método em [34] teve que utilizar a 1 <sup>a</sup> , 2 <sup>a</sup> , 3 <sup>a</sup> , 4 <sup>a</sup> e 5 <sup>a</sup> transmissão para enviar 10,000 pacotes. $E_b/N_0 = 1$ dB. . . . .	p. 64
4.11	Número de vezes que o esquema proposto e o método em [34] teve que utilizar a 1 <sup>a</sup> , 2 <sup>a</sup> , 3 <sup>a</sup> , 4 <sup>a</sup> e 5 <sup>a</sup> transmissão para enviar 10,000 pacotes. $E_b/N_0 = 2$ dB. . . . .	p. 65
4.12	Número de vezes que o esquema proposto e o método em [34] teve que utilizar 1 <sup>a</sup> , 2 <sup>a</sup> , 3 <sup>a</sup> , 4 <sup>a</sup> e 5 <sup>a</sup> transmissão para enviar 10,000 pacotes. $E_b/N_0 = 3$ dB. . . . .	p. 65
5.1	Diagrama de Blocos do sistema de transmissão com múltiplas antenas. . . . .	p. 70

5.2	Taxa de erro de frame (FER) por $E_b/N_0$ para um código turbo de taxa 1/3 com uma, duas e quatro antenas transmissoras usando mapeamento randômico do sinal. Na curva também é mostrado a FER para o caso de uma antena transmissora e um código convolucional de taxa 1/3. . . . .	p. 72
5.3	Taxa de erro de frame (FER) depois da primeira e da segunda transmissões por $E_b/N_0$ para um código turbo de taxa 1/3 com uma e duas antenas transmissoras usando mapeamento randômico do sinal. . . . .	p. 73
5.4	Número total transmissões para comunicação com erro zero de 100,000 pacotes, para o caso de um código turbo de taxa 1/3 com uma, duas e quatro antenas transmissoras usando mapeamento randômico do sinal. Na figura também é mostrado o número total de transmissões para o caso de uma antena transmissora e de um código convolucional de taxa 1/3. Com o valor de $E_b/N_0 = 15$ dB para os quatro casos mostrados na figura. . . . .	p. 74
5.5	Topologia da simulação. . . . .	p. 76
5.6	Throughput do TCP sobre quatro diferentes configurações de links sem fio, considerando TCP New Reno. . . . .	p. 76
5.7	Throughput do TCP sobre quatro diferentes configurações de links sem fio, considerando TCP Reno. . . . .	p. 77
5.8	Throughput do TCP sobre quatro diferentes configurações de links sem fio, considerando TCP Vegas. . . . .	p. 77

# *Lista de Tabelas*

3.1	Tabela de escrita da informação na memória para o interleaver linha-coluna. . . . .	p. 41
3.2	Tabela de leitura da informação na memória para o interleaver linha-coluna. . . . .	p. 41
4.1	Resumo das estratégias de codificação, de transmissão e decodificação para o esquema proposto. . . . .	p. 57
4.2	Número de vezes que o esquema proposto e o método proposto em [34] teve que utilizar a primeira, segunda, terceira, quarta e quinta transmissão, para enviar 10000 pacotes para o receptor. $E_b/N_0 = -1$ dB. . . . .	p. 63
4.3	Número de vezes que o esquema proposto e o método proposto em [34] teve que utilizar a primeira, segunda, terceira, quarta e quinta transmissão, para enviar 10,000 pacotes para o receptor. $E_b/N_0 = 0$ dB. . . . .	p. 66
4.4	Número de vezes que o esquema proposto e o método proposto em [34] teve que utilizar a primeira, segunda, terceira, quarta e quinta transmissão, para enviar 10,000 pacotes para o receptor. $E_b/N_0 = 1$ dB. . . . .	p. 66
4.5	Número de vezes que o esquema proposto e o método proposto em [34] teve que utilizar a primeira, segunda, terceira, quarta e quinta transmissão, para enviar 10,000 pacotes para o receptor. $E_b/N_0 = 2$ dB. . . . .	p. 66
4.6	Número de vezes que o esquema proposto e o método proposto em [34] teve que utilizar a primeira, segunda, terceira, quarta e quinta transmissão, para enviar 10,000 pacotes para o receptor. $E_b/N_0 = 3$ dB. . . . .	p. 66
4.7	Complexidade de decodificação total, definida como em (4.5) e (4.6), para a decodificação de 10,000 pacotes, para o esquema proposto e o método em [34]. . . . .	p. 67
5.1	Valor médio do Throughput das quatro configurações para o TCP Reno e Vegas. . . . .	p. 78
5.2	Taxa de perda para os nós N1, N2, N3 e N4, considerando a implementação do TCP tipo Reno e diferentes configurações na camada física. . . . .	p. 78

- 5.3 Taxa de perda para os nós N1, N2, N3 e N4, considerando a implementação do TCP New Reno e diferentes configurações na camada física. . . . . p. 79
- 5.4 Taxa de perda para os nós N1, N2, N3 e N4, considerando diferentes a implementação do TCP Vegas e diferentes configurações na camada física. . . . . p. 79

# *Lista de Abreviaturas*

<i>BER</i>	Taxa de erro de Bit
<i>FER</i>	Taxa de erro de Frame
<i>ARQ</i>	Automatic Request Repeat
<i>FEC</i>	Forward Error Correction
<i>HARQ</i>	Hybrid ARQ
<i>AWGN</i>	Additive White Gaussian Noise
<i>MIMO</i>	Multiple-Input Multiple-Output
<i>MISO</i>	Multiple-Input Single-Output
<i>SIMO</i>	Single-Input Multiple-Output
<i>SISO</i>	Single-Input Single-Output
<i>MAP</i>	Maximum a Posteriori
<i>RSM</i>	Mapeamento Aleatório do Sinal
<i>TCP</i>	Transport Control Protocol
<i>PSK</i>	Phase Shift Keying
<i>BPSK</i>	Binary Phase Shift Keying
<i>PAM</i>	Pulse Amplitude Modulation
<i>QAM</i>	Quadrature Amplitude Modulation
<i>STC</i>	Space-Time Coding
<i>STTC</i>	Space-Time Trellis Codes
<i>STBC</i>	Space-Time Block Codes
<i>ACK</i>	Acknowledgment

<i>NACK</i>	Negative Acknowledgment
<i>RCPC</i>	Rate Compatible Punctured Convolutional Codes
<i>TCP</i>	Transport Control Protocol
<i>SNR</i>	Relação Sinal Ruído
<i>ICC</i>	International Conference on Communication
<i>RSC</i>	Codificador Convolutacional Recursivo Sistemático
<i>CRC</i>	Cyclic Redundacy Check
<i>LLR</i>	Log Likelihood Ratio
<i>PCCC</i>	Concatenação Paralela dos Códigos Convolutacionais
<i>SCCC</i>	Concatenação Serial dos Códigos Convolutacionais
<i>RCC</i>	Codificador Convolutacional Recursivo
<i>FTP</i>	File Transfer Protocol

# *Resumo*

Este trabalho apresenta dois esquemas de ARQ híbrido utilizando o princípio da codificação turbo. O primeiro esquema, foi proposto para o canal SISO, e o segundo esquema para o Canal MIMO. O primeiro esquema de ARQ híbrido utiliza códigos convolucionais recursivos e o princípio da decodificação turbo. Na primeira transmissão os dados são codificados por um codificador convolucional sistemático recursivo. Na segunda transmissão (primeira retransmissão) os dados são entrelaçados ou embaralhados por um *interleaver* e codificados por um codificador convolucional recursivo. Depois da segunda transmissão o esquema proposto opera como um decodificador turbo. O método proposto é comparado com um esquema conhecido na literatura, ambos em termos do desempenho da taxa de erro, *throughput* e complexidade de decodificação. Foi considerado que ambos os métodos gastam a mesma energia para transmitir um determinado número de bits de informação. O segundo esquema de ARQ híbrido para canal MIMO, utiliza a codificação turbo tradicional e uma técnica simples de diversidade de transmissão, o mapeamento randômico do sinal (RSM). O esquema foi avaliado com diferentes números de antenas transmissoras em relação à taxa de erro, número de retransmissões necessárias e por último, avaliamos como o esquema proposto na camada física pode afetar o *throughput* do Protocolo da Camada de Transporte (TCP).

# *Abstract*

This work shows two hybrid ARQ schemes using the turbo code principle. The first scheme was proposed for the SISO channel, and the second scheme for the MIMO channel. The first hybrid ARQ scheme is based on recursive convolutional codes and turbo decoding principle. In the first transmission the system encodes the data through a systematic recursive convolutional encoder. In the second transmission (first retransmission) data is interleaved prior to being encode by a recursive convolutional encoder. Then, after the second transmission, the scheme operates as a turbo decoder. The proposed method is compared with a popular scheme known in the literature, both in terms of error rate, throughput and decoding complexity. We consider that both schemes spend the same energy for transmission. The second hybrid ARQ scheme for MIMO channel, uses standard parallel turbo code and one simple technique of transmission diversity, random signal mapping (RSM). This scheme was evaluated with different numbers of transmitter antennas in terms of error rate , number of necessary retransmissions and finally, we evaluate how the impact of the proposed scheme in the physical layer can afect the throughput at the Transmission Control Protocol (TCP) in the higher layer.



# 1 *Introdução*

No decorrer de toda a história da civilização as pessoas utilizaram diferentes formas para comunicar-se através de longas distâncias. Por exemplo, o império Romano construiu uma rede de comunicação baseada em sinais de fumaça que alcançavam mais de 4.500 quilômetros [1]. Durante a guerra de Napoleão o telégrafo ótico foi inventado. Desde o século XIX podemos notar o rápido desenvolvimento de novas tecnologias. Todos os anos haviam novas invenções. Começando com o telégrafo elétrico inventado por Samuel Morse em 1844. Em 1896 Guglielmo Marconi inventou o rádio, o primeiro sistema prático de telegrafia sem fios [2]. Hoje, diversos tipos de sistemas de comunicações estão presentes no nosso dia-a-dia, como redes de computadores, redes de televisão, rede de telefonia fixa e os sistemas de telefonia móvel.

Ultimamente tem ocorrido uma demanda crescente por sistemas de comunicações eficientes e confiáveis, que sejam capazes de operar a uma velocidade cada vez maior. Essa demanda exige dos sistemas de comunicações um aumento da taxa de transmissão associado a uma diminuição da taxa de erros.

Claude Shannon em 1948 com o artigo "A Mathematical Theory of Communication" [3], e em 1949 com o artigo "Communication in Presense of Noise" [4], estabeleceu os conceitos da Teoria da Informação. Shannon demonstrou que com uma codificação apropriada da informação erros introduzidos durante a transmissão podem ser reduzidos a um nível desejado. Em um de seus principais resultados, o Teorema da Capacidade do Canal, definiu um limite fundamental na taxa em que a informação pode ser transmitida corretamente através de um canal ruído.

Após o trabalho de Shannon as técnicas de codificação visando à diminuição da taxa de erros (*codificação de canal*) foram muito pesquisadas resultando em vários algoritmos de codificação e decodificação. Dentre eles podem-se destacar os códigos convolucionais e os códigos de bloco.

Durante os primeiros 45 anos de pesquisa, acreditava-se que nenhum sistema de codificação relativamente simples poderia aproximar-se da capacidade do canal. A maior descoberta ocor-

reu em 1993, quando Berrou, Glavieux, e Thitimajshima apresentaram os códigos turbo [5], cujo desempenho em termos de taxa de erro de bit (BER) chegou próxima ao limite de Shannon. Para o código de taxa  $1/2$ , obteve-se uma relação sinal ruído  $E_b/N_0$ , distante apenas 0.7 dB da Capacidade do Canal.

Embora os códigos turbo apresentem um bom desempenho em relação à taxa erro de bit (BER), eles não garantem uma transmissão de dados livre de erros. Quando se deseja um sistema com baixas taxas de erros e alta confiabilidade os códigos corretores podem ser utilizados em conjunto com o protocolo *Automatic-Repeat-Request* (ARQ). Esta combinação das duas técnicas de correção de erros é chamada de ARQ híbrido (HARQ).

Para o canal de transmissão AWGN, os códigos turbo com a mesma complexidade de decodificação que os códigos convolucionais, apresentam desempenho superior. Mas, para o canal *Rayleigh* quase-estático, os códigos turbo não superam o desempenho dos códigos convolucionais com a mesma complexidade de decodificação. Seu desempenho só supera a dos códigos convolucionais quando eles são utilizados em conjunto com mais de uma antena transmissora e/ou receptora [6].

Deste modo, propomos neste trabalho, dois esquemas de ARQ híbrido utilizando o princípio da decodificação turbo. O primeiro esquema para o Canal AWGN SISO, e o segundo esquema para o canal *Rayleigh* MIMO.

## 1.1 Estrutura da Dissertação

Esta dissertação encontra-se organizada da seguinte maneira:

### Capítulo 2 - Modelo do Sistema de Comunicação

Neste capítulo nós apresentamos os princípios básicos de um sistema de comunicação e os modelos matemáticos para os canais de transmissão AWGN e *Rayleigh*. Para o canal *Rayleigh* a técnica de diversidade espacial utilizada para combater os efeitos da atenuação do sinal devido ao desvanecimento é apresentada. Nesta técnica, os sinais são transmitidos e/ou recebidos por múltiplas antenas. Dessa forma, de acordo com o número de antenas transmissoras e receptoras, os modelos matemáticos para os canais SISO, SIMO, MISO e MIMO são apresentados.

E por último, para controlar os erros da transmissão causados pelo ruído adicionado pelo canal as técnicas, *Forward Error Correction* (FEC), *Automatic Repeat Request* (ARQ), e ARQ/FEC híbrido são apresentadas.

### Capítulo 3 - Códigos Turbo

Neste capítulo são apresentados com mais detalhes os códigos turbo, sendo este o tipo de código corretor de erros utilizado no decorrer do trabalho. A estrutura interna do codificador e do decodificador e o algoritmo utilizado para a decodificação, o algoritmo MAP (*maximum a posteriori*) símbolo-a-símbolo [7], são apresentados.

#### Capítulo 4 - Esquema de ARQ híbrido para Canal SISO

Aqui apresentamos um esquema ARQ Híbrido para canal AWGN SISO. Este esquema utiliza códigos convolucionais recursivos e o princípio da decodificação turbo. Na primeira transmissão os dados são codificados por um codificador sistemático recursivo. Na segunda transmissão (primeira retransmissão) os dados são permutados por um *interleaver* e codificados por um codificador recursivo. Depois da segunda transmissão o esquema proposto opera como um decodificador turbo. O método proposto é comparado com um esquema conhecido na literatura.

#### Capítulo 5 - Esquema de ARQ híbrido para Canal MIMO

Neste capítulo, um esquema ARQ híbrido para canal *Rayleigh* com um sistema MIMO é proposto. Neste esquema a diversidade espacial é obtida através de múltiplas antenas transmissoras. Diversidade espacial é alcançada através do mapeamento randômico do sinal (RSM). O ganho de codificação é obtido através da codificação turbo. Resultados numéricos da taxa de erro de frame (FER), número de retransmissões necessárias e o impacto do esquema proposto na performance do protocolo TCP em uma camada superior, variando o número de antenas transmissoras, também é avaliado.

#### Capítulo 6 - Conclusão

Finalmente, neste capítulo revisamos os principais resultados deste trabalho.

É importante salientar que esta dissertação contém material produzido no decorrer deste trabalho, como o Capítulo 5 que foi um artigo publicado e o Capítulo 4 que foi um artigo submetido recentemente. Consequentemente, em cada capítulo desta dissertação há uma notação própria para os símbolos utilizados. Além disso, para cada esquema proposto, o modelo do sistema foi especificado em seus respectivos capítulos.

## 1.2 Contribuições

As principais contribuições encontram-se nos capítulos 4 e 5 da dissertação, e estão brevemente descritas nesta seção.

No Capítulo 4 o novo esquema de ARQ híbrido proposto é comparado com um esquema conhecido na literatura, sendo que ambos os esquemas gastam a mesma energia por uso do canal. Neste capítulo, as principais contribuições são que o esquema proposto alcança um melhor desempenho em termos da taxa de erro do que o outro esquema quando mais de uma transmissão é necessária e ele apresenta uma complexidade de decodificação menor tanto para alta e baixa relação sinal ruído.

E no Capítulo 5 com o outro esquema ARQ híbrido proposto, melhoras significativas em relação a FER e *throughput* podem ser obtidas com o aumento do número de antenas transmissoras. Neste esquema não ocorre o aumento da complexidade ou redução da taxa de transmissão típico dos outros sistemas ARQ espaço-temporais. Além disso, a utilização da codificação e da diversidade através da codificação turbo e do RSM, podem consideravelmente melhorar o desempenho do TCP sobre os links sem fio.

## 2 *Modelo do Sistema de Comunicação*

Este capítulo aborda os principais fundamentos teóricos utilizados nesta dissertação. O capítulo encontra-se organizado da seguinte maneira: Na Seção 2.1 apresentamos o modelo geral de um sistema de comunicação com os principais elementos constituintes. Na Seção 2.2 os canais de transmissão são representados por modelos matemáticos. Na Seção 2.3 apresentamos a técnica de diversidade espacial utilizada nos canais de transmissão sem fio. E na Seção 2.4 as técnicas de controle de erros são apresentadas. Finalmente na Seção 2.5 alguns comentários finais.

### 2.1 **Modelo do Sistema**

O modelo do sistema de comunicação geral utilizado neste trabalho pode ser representado pelo diagrama de blocos da Figura 2.1. O sistema consiste dos seguintes elementos:

A *fonte de informação*, a qual gera a sequência de símbolos de informação. Neste trabalho a fonte de informação utilizada foi a fonte binária equiprovável. Os bits são agrupados em blocos de tamanho  $k$ , denotados de  $u$ .

O *codificador de canal* transforma a sequência de informação  $u$  em uma sequência codificada  $v$ , chamada de palavra-código. O propósito do codificador de canal é introduzir na sequência de informação, de maneira controlada, uma determinada quantidade de informação redundante. Desta forma, o receptor pode utilizar esta informação redundante, para detectar e corrigir erros decorrentes do ruído, quando esta informação é transmitida através de um canal ruidoso.

Os bits codificados  $\{0, 1\}$  da palavra-código, são mapeados para valores reais  $\{-1, +1\}$  de acordo com a modulação BPSK, resultando na sequência  $x$ . Esta sequência é então transmitida através do *canal de transmissão*.

O canal de transmissão representa o meio físico onde o sinal é enviado entre o transmissor e o receptor. Como exemplos de canais de transmissão podemos citar: o ar, através da qual

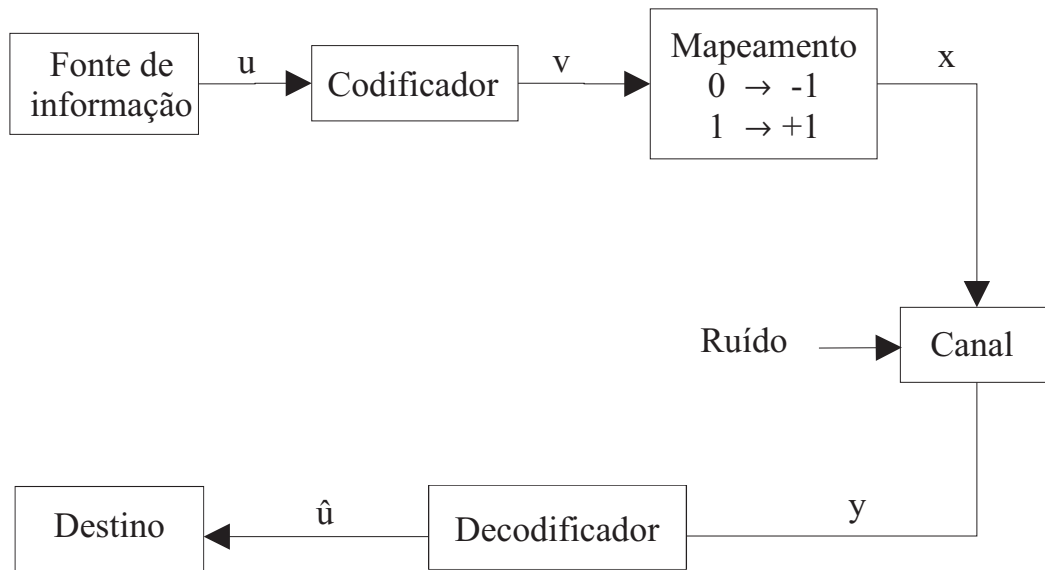


Figura 2.1: Diagrama de Blocos do Modelo do Sistema de Comunicação.

podemos enviar ondas eletromagnéticas (por exemplo ondas de rádio, televisão), por meio de fios ou cabos (como exemplo o telefone), a fibra óptica, a água (como exemplo, sonar) e o vácuo (sondas, naves espaciais, etc..).

O *decodificador de canal* tenta reconstruir a sequência de informação original, de forma que ela apresente a menor taxa de erro possível, devido ao ruído introduzido no canal. Depois da decodificação uma decisão abrupta é tomada para cada símbolo, e a sequência  $\hat{u}$  é entregue ao destino.

Na decisão abrupta, os símbolos são quantizados com um bit de precisão. Se os símbolos forem quantizados com mais de um bit de precisão, a saída é chamada de saída suave.

## 2.2 Modelos Matemáticos para os Canais de Comunicação

No projeto dos sistemas de comunicação, é conveniente construir modelos matemáticos, os quais refletem as características mais importantes do meio de transmissão. Descrevemos dois modelos de canais que são usados para caracterizar os canais físicos que utilizamos neste trabalho.

### 2.2.1 Canal AWGN

O modelo mais simples para o canal de comunicação é o AWGN, o que implica apenas em ruído aditivo no canal. Neste modelo o sinal  $x$  é corrompido por um ruído Gaussiano aditivo  $n$ , o sinal recebido  $y$  é dado pela equação abaixo, como mostra a Figura 2.2.

$$y = x + n \quad (2.1)$$

Fisicamente o ruído aditivo pode ser originário dos componentes eletrônicos e dos amplificadores no receptor do sistema de comunicação, ou de interferência presente no meio.

Se o ruído for introduzido por componentes eletrônicos e amplificadores no receptor, ele pode ser caracterizado como ruído térmico. Este tipo de ruído é caracterizado estatisticamente como um ruído *Gaussiano*. Conseqüentemente, o modelo matemático resultante para o canal é chamado geralmente de *ruído aditivo Gaussiano* [8].

O termo ruído branco se deve ao fato desse tipo de ruído possuir um modelamento matemático que o considera como possuindo largura de faixa infinita, média nula e correlação nula entre suas amplitudes tomadas a instantes de tempos distintos. Ou seja, o valor da amplitude do ruído em um determinado instante independe daquele observado em outro instante de tempo qualquer [9].

Este modelo de canal é razoavelmente preciso em alguns casos, como na comunicação espacial e em algumas comunicações com fio [10].

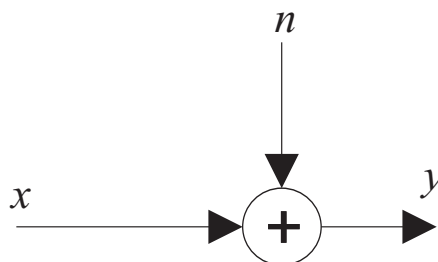


Figura 2.2: Canal AWGN.

### 2.2.2 Canal Rayleigh

A propagação do sinal através do canal sem fio chega ao destino por meio de diferentes caminhos, chamados de multipercursos [11]. Estes multipercursos são formados pela reflexão

e/ou difração e/ou espalhamento do sinal transmitido em estruturas próximas ao receptor, como edifícios, morros, árvores, etc.

A soma vetorial dos vários sinais dos multipercursos pode resultar em uma interferência construtiva ou destrutiva do sinal recebido, resultando assim numa variação da potência do sinal. Esta alteração na intensidade do sinal é denominado de *fading* ou desvanecimento. O termo *Rayleigh* deve-se pelo fato deste fenômeno comportar-se de acordo com a distribuição de probabilidade de *Rayleigh* [8].

Os efeitos do desvanecimento em ambientes móveis, como por exemplo, um usuário de telefone celular caminhando por uma rua urbana, podem ser classificados em dois tipos: rápidos ou lentos. O desvanecimento rápido ocorre quando há uma rápida variação da potência do sinal em um intervalo de tempo curto. No desvanecimento lento a variação da potência do sinal ocorre em um intervalo de tempo maior.

Assim, devido à variação da potência do sinal nos canais sem fio, o sinal recebido  $y$  pode ser dado pela equação (2.2), onde  $h$  é uma componente multiplicativa presente no canal. A Figura 2.3 ilustra este modelo de canal.

$$y = hx + n \quad (2.2)$$

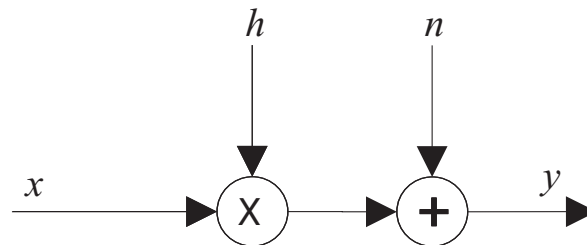


Figura 2.3: Canal Rayleigh.

## 2.3 Diversidade Espacial

Nos canais sem fio ocorre a atenuação do sinal devido ao desvanecimento e interferência entre os usuários, diminuindo desta forma a performance e confiabilidade da comunicação. Para combater estes efeitos pode ser utilizada a técnica da diversidade, na qual diversas réplicas da informação são transmitidas de forma independente. Assim, é alta a probabilidade de que pelo menos uma ou mais destas réplicas não estejam com desvanecimento profundo em determinado instante [12].



São três as principais formas de diversidade utilizadas em canais sem fio:

- Diversidade temporal: Os sinais são transmitidos em tempos diferentes
- Diversidade de frequência: Os sinais são transmitidos em frequências diferentes.
- Diversidade espacial: Os sinais são transmitidos e/ou recebidos por múltiplas antenas.

Entretanto as duas primeiras apresentam alguns inconvenientes. Na diversidade temporal um grande atraso pode ser introduzido, já a diversidade de frequência requer uma largura de banda extra.

Na diversidade espacial podem ser utilizadas várias antenas na transmissão e/ou recepção. A utilização de várias antenas na recepção apresenta algumas desvantagens como custo, tamanho e a potência nas unidades remotas. Consequentemente, as técnicas de diversidade estão quase exclusivamente sendo aplicadas nas estações base para melhorar a qualidade do sinal recebido. Por esta razão a diversidade de transmissão é muito atrativa [13].

Uma maneira simples para explorar a diversidade espacial seria, por exemplo, transmitir o mesmo sinal simultaneamente por mais de uma antena transmissora, entretanto esta técnica simples não resulta em ganho de diversidade, é necessário antes um pré-processamento do sinal transmitido [11].

Várias técnicas foram propostas para incluir diversidade espacial na transmissão. Tarokh *et al*, propuseram em [14] os chamados *space-time trellis codes* (STTC), os quais combinam técnicas de codificação apropriada para múltiplas antenas e processamento do sinal no receptor. Para um número de antenas transmissoras fixo, a complexidade dos STTC (medido pelo número de estados da treliça no decodificador) cresce exponencialmente como função da diversidade e da taxa de transmissão.

Por sua vez, Alamouti em [13] propôs um esquema simples de transmissão usando duas antenas transmissoras. Em [15] os autores introduziram os chamados *space-time block codes* (STBC), onde o esquema de Alamouti é generalizado para um número arbitrário de antenas transmissoras, mantendo a propriedade do processamento linear do sinal no receptor.

Entretanto, o esquema proposto em [15] alcança a máxima taxa de transmissão para um número qualquer de antenas utilizando constelação de sinal real como PAM, por exemplo. Para constelação complexa como PSK ou QAM, ele alcança em geral 1/2 da máxima taxa de transmissão. Para os casos específicos de duas, três e quatro antenas transmissoras e constelação

complexa, o STBC pode alcançar 3/4 da máxima taxa de transmissão. Em [16] os autores propuseram dois esquemas de codificação, com 5 e 6 antenas transmissoras, e taxas de transmissão de 7/11 e 3/5 respectivamente.

Além da redução de taxa de transmissão, que pode ser observado pelos esquemas citados acima, os STBC's requerem uma boa estimação de canal, pois devido à perda de ortogonalidade no receptor, interferência intersimbólica é introduzida diminuindo desta maneira a performance do código.

Neste trabalho, utilizamos uma técnica simples de diversidade de transmissão baseada na codificação de canal tradicional e no mapeamento randômico do sinal (RSM) proposto em [17]. O esquema é apresentado mais detalhadamente no Capítulo 5. Comparado com outros esquemas, o RSM é mais simples e mais flexível ao aumento de antenas transmissoras e/ou receptoras, e principalmente, não reduz a taxa de transmissão do sinal ao adicionar mais antenas, como ocorrem nos esquemas STBCs ortogonais.

### 2.3.1 Modelos Matemáticos dos Canais SISO, SIMO, MISO e MIMO

De acordo com o número de antenas transmissoras e/ou receptoras o canal pode ser classificado como: canal SISO (*single-input single-output*) com uma antena transmissora e uma antena receptora, canal SIMO (*single-input multiple-output*) com uma antena transmissora e múltiplas antenas receptoras, canal MISO (*multiple-input single-output*) com múltiplas antenas transmissoras e uma antena receptora e canal MIMO (*multiple-input multiple-output*) com múltiplas antenas transmissoras e múltiplas antenas receptoras, como mostra a Figura 2.4. Nesta seção nós definimos os modelos matemáticos para estes quatro sistemas.

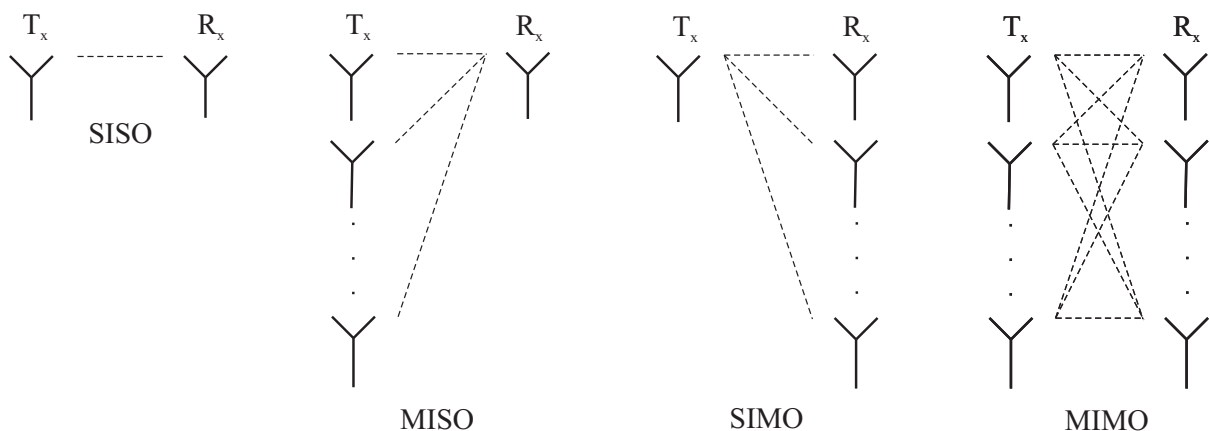


Figura 2.4: Sistemas SISO, MISO, SIMO e MIMO.

## MIMO

Considere o canal MIMO com  $M_T$  antenas transmissoras e  $M_R$  antenas receptoras como na Figura 2.4. O sinal transmitido pela  $j$ -ésima antena transmissora é  $s_j(t)$ . A resposta ao impulso do canal variante no tempo entre a  $j$ -ésima ( $j = 1, 2, \dots, M_T$ ) antena transmissora e  $i$ -ésima ( $i = 1, 2, \dots, M_R$ ) antena receptora é denotado como  $h_{i,j}(\tau, t)$ . A resposta ao impulso para o canal MIMO pode ser expressa como a matriz  $M_T \times M_R$ :

$$H(\tau, t) = \begin{bmatrix} h_{1,1}(\tau, t) & h_{1,2}(\tau, t) & \cdots & h_{1,M_T}(\tau, t) \\ h_{2,1}(\tau, t) & h_{2,2}(\tau, t) & \cdots & h_{2,M_T}(\tau, t) \\ \vdots & \vdots & \ddots & \vdots \\ h_{M_R,1}(\tau, t) & h_{M_R,2}(\tau, t) & \cdots & h_{M_R,M_T}(\tau, t) \end{bmatrix} \quad (2.3)$$

Onde o vetor  $[h_{1,j}(\tau, t) \ h_{2,j}(\tau, t) \ \dots \ h_{M_R,j}(\tau, t)]^T$  é o canal induzido pela  $j$ -ésima antena transmissora através do vetor das antenas receptoras.

O sinal recebido pela  $i$ -ésima antena receptora é dado por:

$$y_i(t) = \sum_{j=1}^{M_T} h_{i,j}(\tau, t) * s_j(t) + n_i(t), \quad i=1, 2, \dots, M_R \quad (2.4)$$

onde  $*$  denota a convolução e  $n_i(t)$  o ruído adicionado pelo canal de transmissão.

## MISO

Se o número de antenas receptoras for reduzido para 1, é obtido um sistema MISO. A resposta ao impulso do canal MISO entre as  $M_T$  antenas transmissoras e a antena receptora pode ser representada por um vetor:

$$\bar{h}(\tau, t) = [h_1(\tau, t) \ h_2(\tau, t) \ \dots \ h_{M_T}(\tau, t)] \quad (2.5)$$

O sinal recebido pode ser expresso como:

$$y(t) = \sum_{j=1}^{M_T} h_j(\tau, t) * s_j(t) + n(t) \quad (2.6)$$

a qual pode ser expressa pela notação vetorial:

$$y(t) = \bar{h}_j(\tau, t) * \bar{s}_j(t) + n(t) \quad (2.7)$$

onde o vetor do sinal de entrada é o vetor  $\bar{s}(t) = [s_1(t) \quad s_2(t) \quad \dots \quad s_{M_T}(t)]^T$  de dimensão  $M_T \times 1$ .

## SIMO

Se o número de antenas transmissoras for reduzido para 1 e o número de antenas receptoras for  $M_R$  o sistema é chamado de SIMO. O resposta ao impulso do canal SIMO entre a antena transmissora e as  $M_R$  antenas receptoras pode ser representada por um vetor:

$$\bar{h}(\tau, t) = [h_1(\tau, t) \quad h_2(\tau, t) \quad \dots \quad h_{M_R}(\tau, t)]^T \quad (2.8)$$

O sinal recebido na  $i$ -ésima antena receptora pode ser escrito como:

$$y_i(t) = h_i(\tau, t) * s(t) + n_i(t) \quad (2.9)$$

ou na forma vetorial  $\bar{y}(t) = [y_1(t) \quad y_2(t) \quad \dots \quad y_{M_R}(t)]^T$  por:

$$\bar{y}(t) = \bar{h}(\tau, t) * \bar{s}(t) + n(t) \quad (2.10)$$

## SISO

Se o número de antenas receptoras e transmissoras for reduzido para 1, é obtido um sistema SISO. O sinal recebido  $y(t)$  é dado por:

$$y(t) = h(\tau, t) * s(t) + n(t), i = 1, 2, \dots, M_R \quad (2.11)$$

onde  $n(t)$  é o ruído adicionado pelo canal de transmissão.

Se o canal for com desvanecimento rápido,  $h(\tau, t)$  varia para cada transmissão de  $s(t)$ , se no entanto, o canal for desvanecimento lento, usamos o mesmo  $h(\tau, t)$  para o bloco de informação  $s(t)$ .

### 2.3.2 Efeito da Diversidade Espacial

A Figura 2.5 mostra a variação na potência do sinal em função do tempo quando a técnica da diversidade espacial é utilizada. Esta simulação foi feita no Matlab, sendo que o canal efetivo foi simulado utilizando o quadrado da norma de Frobenius  $\|H\|_F^2$ , como em [11].

Pela figura podemos notar que, onde a técnica da diversidade espacial não é utilizada, canal SISO com  $M_T = M_R = 1$ , o sinal sofre com algumas variações profundas da potência do sinal. Quando a técnica da diversidade espacial é utilizada, canal MIMO com  $M_T = M_R = 2$ , verifica-se que ocorre uma diminuição da variação da potência do sinal. Com  $M_T = M_R = 6$ , além da diminuição da variação da potência do sinal, verifica-se também um aumento na média da potência do sinal.

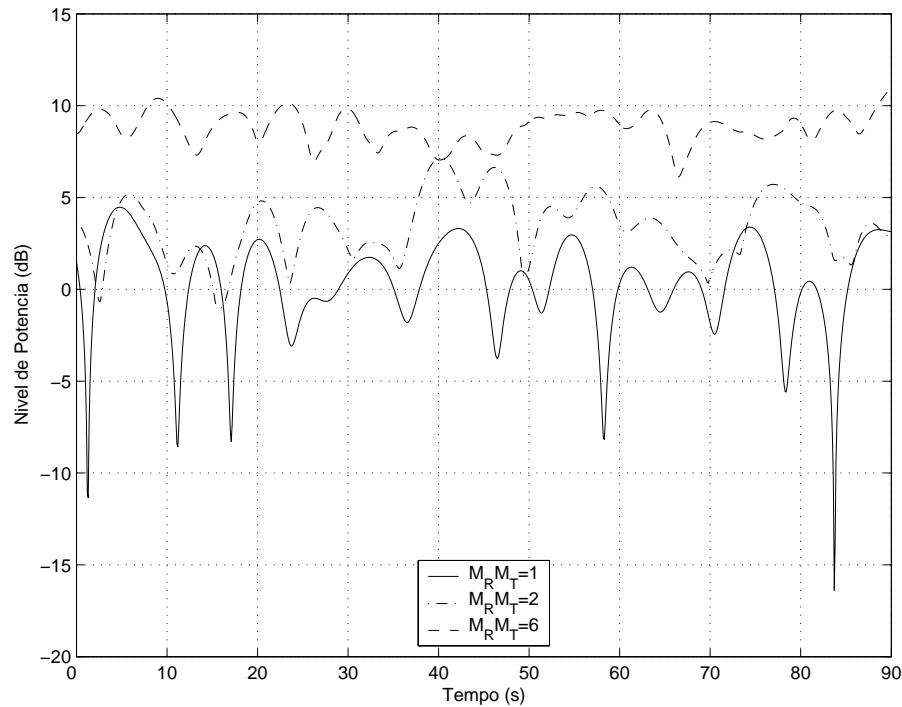


Figura 2.5: Efeito do aumento do número de antenas na variação do nível da potência do sinal recebido.

## 2.4 Técnicas Fundamentais de Controle de Erros

Um dos interesses principais na transmissão de dados é como controlar os erros da transmissão causados pelo ruído do canal de modo que os dados possam ser entregues ao usuário sem erros [18]. Existem três categorias de esquemas para controle de erros [18]: *Forward Error Correction* (FEC), *Automatic Repeat Request* (ARQ), e o esquema ARQ/FEC híbrido.

### 2.4.1 FEC

No sistema de controle de erro FEC, códigos corretores de erros são utilizados para detectar e corrigir os erros. Neste esquema, bits de redundância são acrescentados à informação para formar a palavra-código. Quando o receptor detectar erros na palavra recebida, o código

corretor tenta localizar e corrigir os erros. Depois que a correção de erros é feita, a palavra decodificada (com os bits de redundância removidos) é entregue ao usuário.

Existem vários tipos de códigos corretores de erros. Entre eles, podemos citar os códigos de bloco, códigos convolucionais e os códigos turbo.

### **Códigos de bloco**

Um código de bloco utiliza  $k$  bits de informação gerados pela fonte para gerar uma palavra-código com  $n$  bits, acrescentando assim,  $n - k$  bits de redundância. A relação entre o número de bits de um bloco de informação e o número de bits da palavra-código correspondente,  $k/n$ , é a taxa de codificação. De maneira geral, quanto menor a taxa de codificação, maior é a capacidade de detecção e correção de erros [9].

Além da taxa do código, outro parâmetro importante é a distância de Hamming e o peso da palavra-código. A distância da palavra-código é definida como o número de elementos nos quais duas palavras-códigos  $C_i$  e  $C_j$  diferem. A distância mínima é definida como a menor distância de Hamming para um dado conjunto de palavras-código.

O peso da palavra-código é dado pelo número de elementos diferentes de zero. Para o código binário, o peso é basicamente o número de 1's na palavra-código.

### **Códigos Convolucionais**

Os códigos convolucionais são chamados de códigos com memória, ou seja, a codificação da informação depende dos blocos anteriores. Um código convolucional  $(n, k, m)$  gera  $n$  bits codificados para  $k$  bits de entrada, onde  $m$  é a memória do codificador. As  $n$  saídas do codificador, em um dado instante, dependem não apenas das  $k$  entradas mas também das  $m$  entradas anteriores [19].

Seja o exemplo da Figura 2.6 dado em [19], de um codificador convolucional de taxa 1/2 e  $m = 4$ . Para cada bit de entrada, há dois bits de saída os quais dependem dos 4 bits de entrada anteriores. Dada a sequência de entrada  $u = (u_0, u_1, u_2, \dots)$  as duas sequências de saída do codificador  $v^{(1)} = (v_0^{(1)}, v_1^{(1)}, v_2^{(1)}, \dots)$  e  $v^{(2)} = (v_0^{(2)}, v_1^{(2)}, v_2^{(2)}, \dots)$  são iguais à convolução da sequência de entrada  $u$  com as duas sequências geradoras  $g^{(1)} = (1, 0, 0, 1, 1)$  e  $g^{(2)} = (1, 1, 1, 0, 1)$ . Portanto, as equações de codificação são:  $v^{(1)} = u * g^{(1)}$  e  $v^{(2)} = u * g^{(2)}$ .

Em geral, para códigos convolucionais cada sequência de informação (entrada, saída ou sequência geradora) pode ser escrita na forma polinomial, com os coeficientes do polinômio

igual aos elementos da sequência correspondente. Neste caso, a saída polinomial é igual ao produto polinomial da entrada com o polinômio gerador [19].

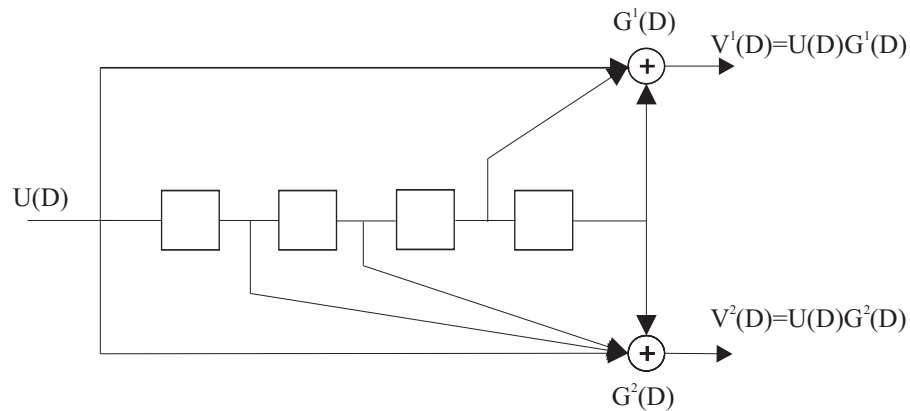


Figura 2.6: Código convolucional de taxa 1/2 e  $m=4$ .

Para o exemplo dado em [19] as equações de codificação são:

$$V^{(i)}(D) = U(D)G^{(i)}(D), \quad i=1,2 \quad (2.12)$$

onde

$$\begin{aligned} U(D) &= u_0 + u_1D + u_2D^2 + \dots \\ V^{(1)}(D) &= v_0^{(1)} + v_1^{(1)}D + v_2^{(1)}D^2 + \dots \\ V^{(2)}(D) &= v_0^{(2)} + v_1^{(2)}D + v_2^{(2)}D^2 + \dots \end{aligned}$$

e

$$\begin{aligned} G^{(1)}(D) &= 1 + D + D^4, \\ G^{(2)}(D) &= 1 + D^2 + D^3 + D^4, \end{aligned}$$

Existem várias técnicas para decodificar os códigos convolucionais. O mais importante destes métodos é o algoritmo de Viterbi, o qual executa a decodificação por máxima verossimilhança dos códigos convolucionais [20].

### Códigos Turbo

Utilizamos neste trabalho um código turbo [5] como código corretor de erro, o qual será explicado com mais detalhes no Capítulo 3. A escolha deste tipo de código corretor de erro se

deve ao fato de seu excelente desempenho. Seu desempenho em termos da taxa de erro de bit (BER) chega muito perto da capacidade de Shannon [4].

O codificador turbo é formado basicamente pela concatenação paralela de dois códigos convolucionais recursivos sistemáticos separados por um *interleaver*, podendo também ser formado por mais de dois códigos convolucionais, chamados então de códigos turbo múltiplos [21].

O decodificador turbo tradicional é formado pela concatenação serial de dois decodificadores separados por um *interleaver*, decodificados iterativamente pelo algoritmo BCJR [7] modificado. Neste processo de decodificação, informações sobre a confiabilidade ou qualidade da decodificação de um dos decodificadores alimenta o processo de decodificação do outro decodificador. Dessa forma, a cada iteração tem-se maior confiabilidade na estimação dos bits de informação.

Além disso, em [6] os autores mostraram que no canal AWGN, um código turbo com a mesma complexidade de decodificação que um código convolucional, apresenta desempenho superior. Entretanto, para o canal *Rayleigh* quase-estático (lento), um código turbo não supera o desempenho dos códigos convolucionais com a mesma complexidade de decodificação, seu desempenho só supera o dos códigos convolucionais quando ele é utilizado em conjunto com mais de uma antena transmissora e/ou receptora.

## 2.4.2 ARQ

No sistema de controle de erros FEC, a palavra decodificada é entregue ao usuário estando ela correta ou incorreta. Assim, para aumentar a confiabilidade da transmissão o protocolo ARQ apresenta uma solução alternativa. No protocolo ARQ há somente a capacidade de detecção de erros e nenhuma tentativa de corrigir os pacotes recebidos com erros é feita, ao invés disso é feito um pedido de retransmissão da informação [19].

Neste protocolo, se no receptor não forem detectados erros então a mensagem é entregue ao destinatário. Se o receptor detectar erros na mensagem recebida, é feito um pedido para a retransmissão da mensagem. Se o receptor detectar novamente erros na mensagem recebida outro pedido de retransmissão é feito, o processo continua até que a informação seja recebida corretamente. Com este esquema de controle de erros, a informação é entregue ao usuário incorreta somente se o receptor falhar na detecção de erros. Conseqüente, o esquema ARQ apresenta alta confiabilidade.

Existem dois parâmetros que podem ser usados para avaliar a performance do protocolo ARQ: confiabilidade e *throughput* [22]. A confiabilidade é frequentemente expressa em termos



da porcentagem de pacotes aceitos que contém um ou mais erros.

O *throughput* é o número médio de pacotes aceitos no receptor. Neste protocolo o desempenho do *throughput* depende principalmente do número de vezes que a informação tem que ser transmitida até que seja aceita pelo receptor.

Existem três tipos básicos de protocolos ARQ: *stop-and-wait*, *go-back-N* e *selective-repeat*. Eles estão explicados nas seções abaixo.

### Protocolo ARQ *stop-and-wait*

O protocolo *stop-and-wait* é o mais simples dos três protocolos, neste protocolo o transmissor envia o pacote de dados e espera um reconhecimento (ACK) do receptor. Se for detectado erro o receptor envia um pedido de retransmissão da informação (RQ ou NACK) ou envia um ACK para o transmissor se a informação estiver livre de erro. Este processo está ilustrado na Figura 2.7.

O principal benefício do protocolo *stop-and-wait* é que ele não precisa de memória para o transmissor e receptor [22]. Pela Figura 2.7 vemos que o pacote de informação 2 não é transmitido até que o pacote 1 seja aceito no receptor.

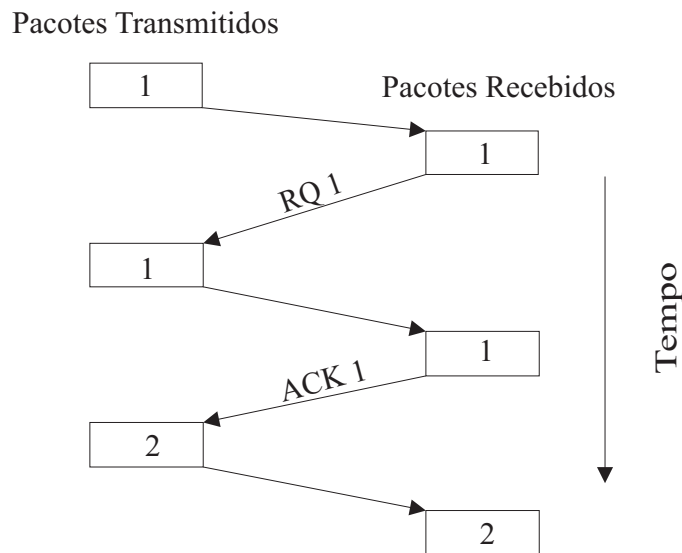


Figura 2.7: Diagrama de tráfego de pacotes do protocolo ARQ *stop-and-wait*.

### Protocolo ARQ *go-back-N*

No protocolo *go-back-N* o transmissor envia pacotes de dados num fluxo contínuo. Quando o receptor detectar erros no pacote recebido, ele envia um pedido de retransmissão da informa-

ção e espera pela segunda cópia. Todos os outros pacotes subsequentes são ignorados até que o segundo pacote seja recebido [22], como mostra a Figura 2.8.

Ignorando os pacotes que seguem um pedido de retransmissão, a memória no receptor é evitada. O transmissor, por outro lado, deve responder reenviando o pacote pedido e todos os pacotes subsequentes, conseqüentemente é necessária memória no transmissor. O termo *go-back-N* deriva o fato de que quando o transmissor recebe um pedido de retransmissão da informação, ele deve voltar na memória  $N$  pacotes e recomençar a transmissão neste ponto [22].

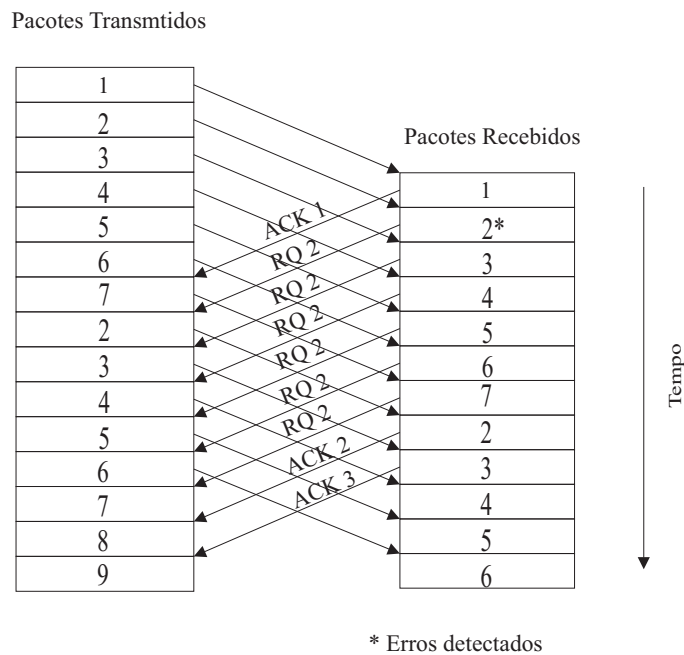


Figura 2.8: Diagrama de tráfego de pacotes do protocolo ARQ *go-back-N*.

### Protocolo ARQ *selective-repeat*

Como no protocolo *go-back-N*, no protocolo *selective-repeat* o transmissor envia pacotes de dados num fluxo contínuo. Entretanto, neste protocolo, quando o emissor recebe um NACK correspondente a uma determinada mensagem, apenas essa mensagem é repetida e não todas as que se lhe seguem. Portanto, só são retransmitidas as mensagens que foram recebidas com erro. No entanto, a complexidade aumenta, já que o receptor necessita neste caso de ter a capacidade de reagrupar as mensagens na ordem correta antes de as enviar para o destinatário. A operação deste tipo de protocolo está ilustrada na Figura 2.9. A utilização deste protocolo apresenta alguns inconvenientes, como o tamanho e custo da memória necessária, e o consumo de potência [22].

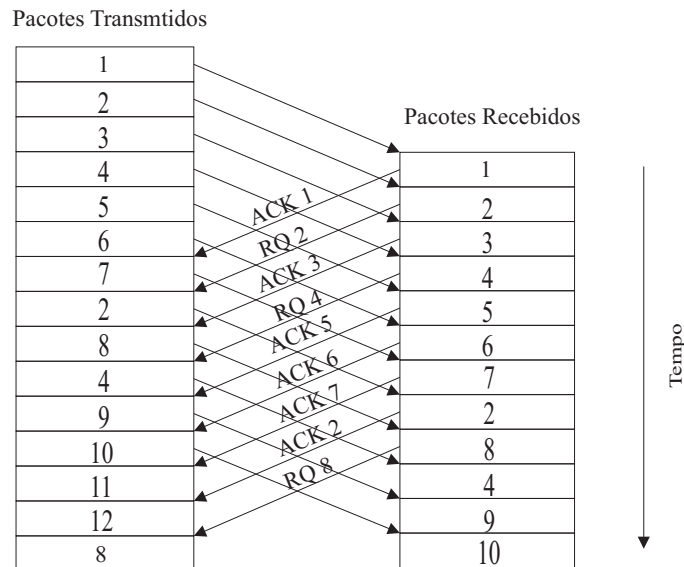


Figura 2.9: Diagrama de tráfego de pacotes do protocolo ARQ *Selective Repeat*.

### 2.4.3 ARQ Híbrido

Quando se deseja um sistema com baixas taxas de erros e alta confiabilidade a técnica de correção de erros FEC é utilizada em conjunto com um protocolo ARQ. Esta combinação dos dois protocolos de correção de erros é chamado de ARQ Híbrido (HARQ) ou ARQ/FEC híbrido.

A função do sistema FEC é reduzir a frequência das transmissões corrigindo os erros, aumentando assim o *throughput* do sistema. Entretanto, quando o sistema FEC não conseguir corrigir os erros, ao invés de entregar a informação incorreta ao usuário, o receptor faz um pedido de retransmissão da informação (ARQ), aumentando assim a confiabilidade do sistema.

Os esquemas ARQ híbridos podem ser classificados como tipo-I e tipo-II. No HARQ tipo-I, se a palavra decodificada estiver errada, os pacotes errados são descartados e o receptor envia um pedido de retransmissão da informação para o transmissor. Neste caso, o pacote inteiro é retransmitido.

O HARQ tipo II, é um protocolo que se adapta às mudanças das condições do canal através do uso de *redundância incremental*. O transmissor neste sistema responde ao pedido de retransmissão enviando bits de paridade adicionais para o receptor. O receptor adiciona estes bits no pacote recebido, aumentando assim a capacidade de correção de erros.

A técnica que utiliza redundância incremental também pode ser chamada de combinação de pacotes. Nesta técnica, o pacote que gerou o pedido de retransmissão da informação é guardado,

e mais tarde é combinado com cópias adicionais dos pacotes, criando um único pacote mais confiável do que qualquer um dos outros pacotes constituintes. Esta idéia foi primeiramente proposta por Sidhu em [23].

Vários outros esquemas foram propostos que envolvem alguma forma de combinação de pacotes. Em [24] várias cópias dos pacotes transmitidos e codificados são combinadas antes da decodificação. Em [25] os autores propuseram alguns esquemas com a utilização da combinação de pacotes em conjunto com o decodificador de Viterbi.

Em [26] os autores propuseram um esquema chamado *Rate Compatible Punctured Convolutional Codes* (RCPC). O princípio deste esquema é não repetir a informação ou os bits de paridade se a transmissão for mal sucedida, mas transmitir bits codificados adicionais a taxas mais baixas até que o código esteja poderoso o suficiente para permitir a decodificação.

## 2.5 Comentários Finais

Neste capítulo apresentamos um modelo geral do sistema de transmissão da informação utilizado neste trabalho. Os modelos matemáticos para os canais de transmissão AWGN e Rayleigh também foram apresentados. Para controlar os erros da transmissão causados pelo ruído adicionado pelo canal, foram apresentadas três técnicas: os códigos corretores de erros (FEC), protocolos de retransmissão da informação (ARQ) e esquemas FEC/ARQ híbridos.

## 3 Códigos Turbo

### 3.1 Introdução

Em 1948 Shannon introduziu o conceito de *Capacidade de Canal* como sendo a máxima taxa que a informação pode ser transmitida através de um canal ruidoso [4]. Em se tratando de comunicação digital em canal AWGN, foi demonstrado que a maior taxa de transmissão que pode trafegar neste canal é dada pela seguinte equação:

$$C = W \log_2 \left( 1 + \frac{E_s}{N_0} \right) \text{ (bits por segundo)} \quad (3.1)$$

onde  $W$  é a largura de banda do canal,  $E_s$  é a energia média em cada intervalo de sinal de duração  $T = 1/W$ , e  $N_0/2$  é a densidade espectral de potência do ruído.

Considere agora a relação a seguir, onde  $\eta$  representa a eficiência espectral.

$$0 < \eta < \frac{C}{W}, \quad (3.2)$$

e

$$\frac{E_s}{N_0} = \eta \frac{E_b}{N_0}, \quad (3.3)$$

Onde  $\eta$  representa o número médio de bits de informação transmitidos por intervalo de símbolo de duração de  $T$  segundos e  $E_b$  é a energia média por bit. Substituindo as equações acima em (3.1), e com algumas manipulações, obtém-se:

$$\frac{E_b}{N_0} > \frac{2^\eta - 1}{\eta} \quad (3.4)$$

a qual relaciona a eficiência espectral com a relação sinal ruído (SNR) representada por  $E_b/N_0$ .

O Teorema da Capacidade de Canal [4] garante a existência de códigos corretores de erro que podem alcançar taxas de erros baixas quanto se queira se a taxa de transmissão for menor do que a capacidade do canal. Este teorema não especificou como projetar um código capaz de alcançar esta taxa, o que motivou o desenvolvimento de várias técnicas de correção de erros.

Durante décadas novos códigos foram propostos para aproximar-se, em desempenho, da Capacidade de Canal. Em 1986 na missão espacial *Voyager* para Urano, foi utilizado o código convolucional (2,1,6) concatenado com o código Reed-Solomon (255,223). Este esquema de concatenação de códigos pode alcançar a taxa de erro bit de  $10^{-5}$  com  $E_b/N_0 = 2.6$  dB e eficiência espectral de 0.4370 bits/seg/Hz. Um esquema mais potente de concatenação de códigos foi escolhido para a nave espacial *Galileo* lançada em 1989. Neste esquema foi feita a concatenação serial do código convolucional (4,1,14) com o código Reed-Solomon (255,233). Para a taxa de erro bit de  $10^{-5}$  este esquema precisa de pelo menos  $E_b/N_0 = 0.8$  dB com eficiência espectral de 0.219 bits/seg/Hz [29].

Com uma abordagem diferente Berrou, Glavieux e Thitimajshima [5] em 1993, apresentaram uma nova classe de códigos corretores de erros denominado *Códigos Turbo*, cujo desempenho em termos de taxa de erro de bit (BER) chegou próxima ao limite de Shannon. Para o código de taxa 1/2, obteve-se uma  $E_b/N_0$  distante apenas 0.7 dB da Capacidade de Shannon.

O trabalho deste grupo de pesquisadores, antes desconhecido da comunidade de codificação, foi apresentado na *International Conference on Communications* (ICC). Resultados parecidos foram submetidos no ano anterior, mas o artigo não foi aceito porque os resultados eram muitos bons para ser verdadeiros [27]. Tendo sido o trabalho publicado, outros pesquisadores puderam comprovar os resultados.

Este novo esquema de codificação de canal é formado pela concatenação paralela de dois códigos convolucionais recursivos sistemáticos, decodificados iterativamente por um algoritmo baseado no algoritmo MAP (*maximum a posteriori*) símbolo-a-símbolo BCJR<sup>1</sup> [7].

Ao contrário de outros decodificadores convencionais, o decodificador turbo consiste em dois decodificadores concatenados serialmente através de um *interleaver*. O primeiro decodificador calcula uma estimativa de probabilidade para cada bit, a qual será a nova probabilidade *a priori* para o segundo decodificador. O segundo decodificador utiliza a probabilidade *a priori* calculada no primeiro decodificador, e calcula uma nova probabilidade para cada bit, chamada de probabilidade extrínseca. A probabilidade extrínseca calculada pelo segundo decodificador é utilizada pelo primeiro decodificador como a nova probabilidade *a priori*. Este processo continua iterativamente até que uma decisão mais confiável é tomada para cada bit.

---

<sup>1</sup>A sigla BCJR foi adotada devido ao nome dos inventores do algoritmo de decodificação: L.R.Bahl, J. Cocke, F. Jelinek e J.Raviv

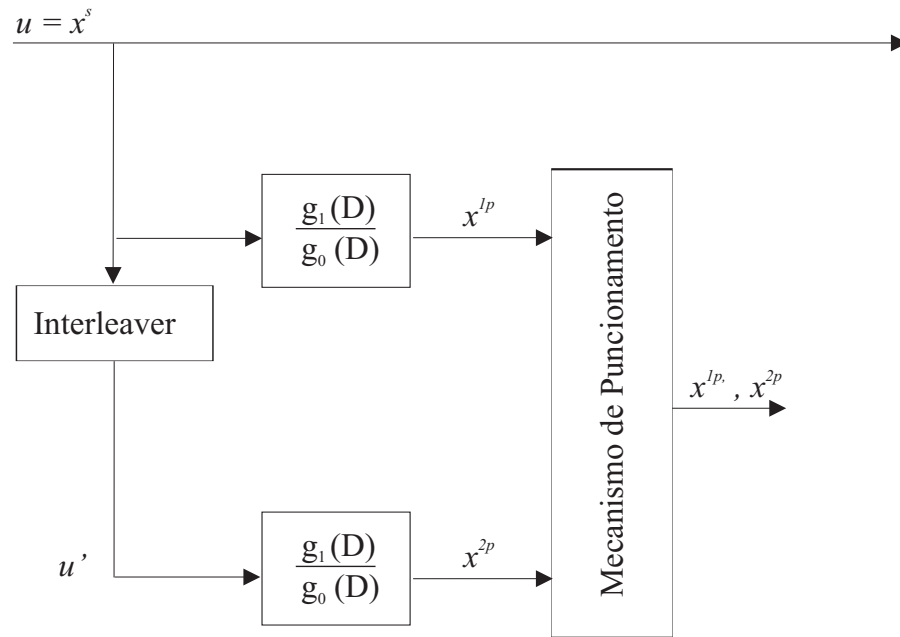


Figura 3.1: Diagrama de Blocos do Codificador Turbo.

## 3.2 Codificador Turbo

A Figura 3.1 mostra a estrutura tradicional do codificador turbo. Conforme apresentado em [5], o codificador turbo consiste em dois codificadores convolucionais sistemáticos recursivos (RSC) separados por um *interleaver* de comprimento  $N$ . Nesta figura também é mostrado o mecanismo de puncionamento, o qual será explicado na seção 3.2.3.

A sequência de informação  $u_k$  é codificada nos dois RSC constituintes do código turbo, mas em ordem diferente. O primeiro codificador opera diretamente com a sequência de entrada. O segundo codificador opera com a entrada  $u'_k$  que é a entrada  $u_k$  permutada pelo *interleaver*. Depois da codificação existem três fluxos de dados, os bits sistemáticos  $x^s$ , que são os bits de entrada e os bits de paridade  $x^{1p}$  e  $x^{2p}$  dos dois codificadores sistemáticos recursivos.

São acrescentados no final de cada sequência de entrada, bits adicionais para forçar o codificador retornar ao estado zero. Estes bits são chamados de bits de cauda ("*tail bits*").

Assim, a taxa do código é definida como  $k/3(k+m)$  onde  $k$  é o número de bits da entrada do codificador e  $m$  o número de bits necessários para o codificador retornar ao estado zero. Para valores grandes de  $k$  a taxa do codificador turbo pode ser aproximada para  $1/3$ .

### 3.2.1 Codificadores Sistemáticos Recursivos

Em [28] os autores avaliaram o desempenho do codificador sistemático recursivo e do codificador não sistemático recursivo como codificadores constituintes do codificador turbo. Para o valor de  $E_b/N_0 = 1.76$  dB a taxa de erro de bit (BER) para o codificador turbo formado pela concatenação do codificador sistemático foi de  $1.0 \times 10^{-5}$ . Para o codificador turbo formado pela concatenação dos códigos convolucionais não sistemáticos a BER obtida foi de  $7.2 \times 10^{-3}$ , ou seja, o seu desempenho foi cerca de três vezes pior em ordem de magnitude do que o codificador sistemático. Consequentemente, codificadores convolucionais não sistemáticos não são uma boa alternativa como codificadores constituintes do codificador turbo para  $u_k$  equiprovável.

Neste trabalho os autores também mostraram que os codificadores sistemáticos geralmente tem performance melhor do que os codificadores não sistemáticos como constituintes do código turbo, para os valores menores de  $E_b/N_0$ , a região de maior interesse. Isto pode ser explicado pelo fato de que os codificadores sistemáticos, ao contrário do que a maioria dos codificadores não sistemáticos, mapeiam sequências de entrada com peso baixo em palavras-códigos com peso baixo também. Para os valores de  $E_b/N_0$  maiores, os codificadores não sistemáticos apresentam um desempenho melhor do que os codificadores sistemáticos, isto se deve ao fato de que eles apresentam distância livre ( $d_{free}$ ) maior do que os sistemáticos.

Já a característica recursiva se deve ao fato de que um bom código linear é aquele que contém na maioria das vezes palavras-códigos com peso de Hamming maior, exceto é claro a palavra-código zero. Palavras códigos com peso alto são desejáveis porque isto significa que elas são mais distintas entre si e assim aumenta a probabilidade do decodificador diferenciá-las.

Se o codificador for não recursivo, uma sequência de entrada com peso baixo resulta na sequência de paridade com peso baixo também. Por outro lado se o codificador for recursivo há uma grande chance que uma sequência de entrada com peso baixo resulte em um sequência de paridade com peso alto.

Por exemplo, seja o codificador não recursivo da Figura 3.2 de taxa 1/2 com polinômios geradores  $g_0 = [11]$  e  $g_1 = [10]$ . Dada a sequência de entrada  $u = (00001000)$  com peso 1, a saída do codificador  $x^p$  tem peso 2. Supondo que a sequência de entrada  $u$  seja permutada pelo *interleaver* mudando a ordem dos bits da sequência para  $u = (10000000)$  a saída do codificador  $x^p$  continua com peso 2. Ou seja, o *interleaver* não influenciou na distribuição do peso desta palavra-código.

Seja o codificador recursivo da Figura 3.3 de taxa 1/2 com polinômios geradores  $g_0 = [111]$  e  $g_1 = [101]$ . Dada a sequência de entrada  $u = (00111000)$ , a saída do codificador  $x^p$  tem



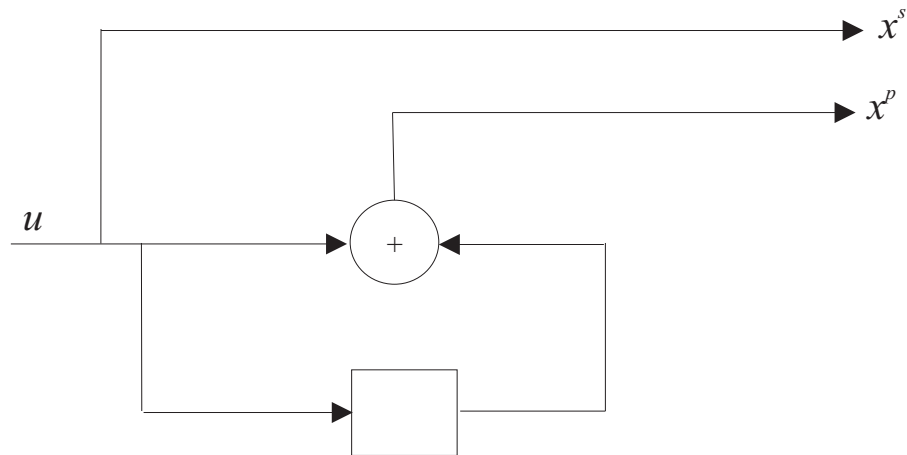


Figura 3.2: Codificador Convolucional Sistemático não Recursivo.

peso 2. Se a sequência de entrada for permutada pelo *interleaver* mudando a ordem dos bits para  $u = (10101000)$  a saída do codificador  $x^p$  terá peso 4. Seja outra sequência de entrada  $u = (00110000)$  com peso 2, a saída do codificador  $x^p$  tem peso 3. Se esta sequência de entrada for permutada pelo *interleaver* mudando a ordem dos bits para  $u = (01000001)$  a saída do codificador  $x^{2p}$  terá peso 5. Assim, se o codificador for recursivo, aumenta a chance de que uma sequência de entrada com peso baixo resulte em um sequência de paridade com peso alto.

### 3.2.2 Interleaver

O peso da palavra-código gerada pelo codificador turbo é a soma do peso da palavra-código dos bits sistemáticos com o peso das palavras-códigos dos bits de paridade da saída dos dois codificadores constituintes. No codificador turbo se uma das saídas dos codificadores tiver peso pequeno a outra provavelmente terá peso maior. Isto se deve ao fato que a entrada do segundo codificador foi permutada pelo *interleaver* e sua saída é geralmente bastante diferente do outro codificador.

Assim é possível que uma das duas saídas dos codificadores produza uma palavra-código com peso pequeno, mas a probabilidade de que os dois codificadores produzam saídas com peso pequeno é extremamente pequena. Esta melhora é chamada de *ganho de interleaver* e é uma das razões porque códigos turbo tem um desempenho tão bom.

A outra finalidade do interleaver é descorrelacionar as entradas dos dois decodificadores. Se as entradas dos dois decodificadores estão descorrelacionadas é alta a probabilidade de que depois de alguma correção de erro em um dos decodificadores alguns dos bits errados restantes sejam corrigidos no outro decodificador [29].

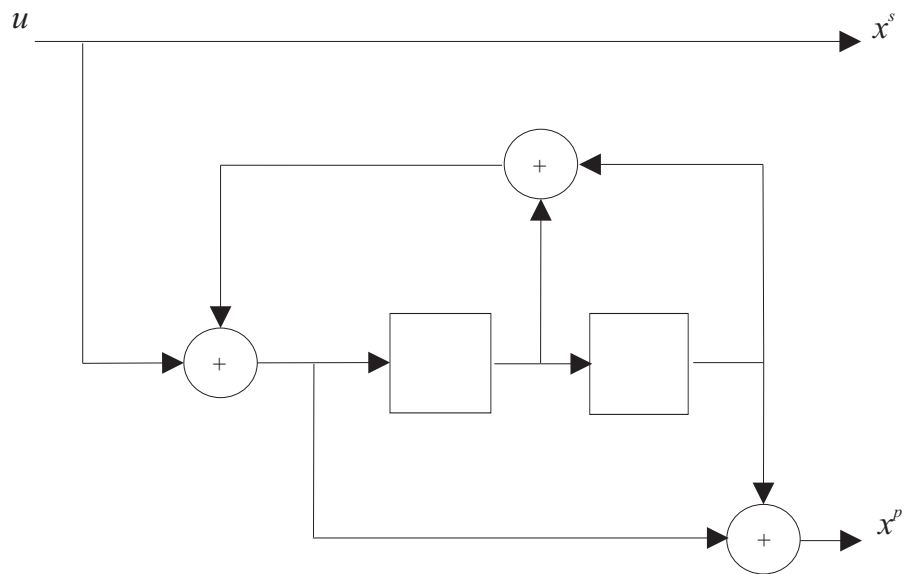


Figura 3.3: Codificador Convolutivo Sistemático Recursivo.

O projeto do *interleaver* é um fator importante, o qual determina o bom desempenho do código turbo. Assim, muitos pesquisadores vêm propondo várias maneiras para permutar a sequência de informação. As seguintes seções mostram alguns tipos de *interleavers* utilizados no código turbo.

### Interleaver de Bloco

Neste tipo de *interleaver* a informação é escrita na memória em linha e é lida por coluna. Por exemplo, seja a sequência de informação de  $x_1$  até  $x_{21}$ , ela é escrita como na Tabela 3.1 e lida como na Tabela 3.2.

$x_1$	$x_2$	$x_3$
$x_4$	$x_5$	$x_6$
$x_7$	$x_8$	$x_9$
$x_{10}$	$x_{11}$	$x_{12}$
$x_{13}$	$x_{14}$	$x_{15}$
$x_{16}$	$x_{17}$	$x_{18}$
$x_{19}$	$x_{20}$	$x_{21}$

Tabela 3.1: Tabela de escrita da informação na memória para o interleaver linha-coluna.

$x_1$	$x_4$	$x_7$	$x_{10}$	$x_{13}$	$x_{16}$	$x_{19}$	$x_2$	$x_5$	$x_8$	$x_{11}$	$x_{14}$	$x_{17}$	$x_{20}$	$x_3$	...
-------	-------	-------	----------	----------	----------	----------	-------	-------	-------	----------	----------	----------	----------	-------	-----

Tabela 3.2: Tabela de leitura da informação na memória para o interleaver linha-coluna.

### Interleaver Pseudo-Aleatório

Os *interleavers* pseudo aleatórios são definidos por um gerador de números pseudo rãndômicos, ou uma tabela de procura com todos os números inteiros de 1 até N, onde N é o tamanho do bloco a ser gerado.

### Interleaver Tipo S

Interleaver tipo S, onde ( $S = 1,2,3\dots$ ) é um *interleaver* semi randômico, construído da seguinte maneira. Cada número inteiro escolhido randômicamente é comparado com S previamente escolhido, se a diferença entre esta seleção e S for menor do que S, os números inteiros são rejeitados. O processo é feito até que N inteiros distintos sejam selecionados .

### 3.2.3 Puncionamento

Taxas de códigos mais altas como 1/2, 2/3, 3/4 e outras, podem ser obtidas através do puncionamento do codificador turbo de taxa 1/3. O processo de puncionamento consiste em deletar os bits de saída do codificador de acordo com a matriz de puncionamento escolhida.

Para um código turbo com taxa 1/2 a matriz de puncionamento comumente usada é dada por:

$$P = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3.5)$$

De acordo com esta matriz de puncionamento os bits de paridade dos dois codificadores constituintes são alternadamente deletados. Desta forma, o símbolo do código turbo em um dado instante, é formado pelo dígito de informação  $x^s$  seguido pelo dígito de paridade  $x^{1p}$  do primeiro codificador ou  $x^{2p}$  do segundo codificador, os quais são obtidos alternadamente.

## 3.3 Decodificador

O decodificador turbo, mostrado na Figura 3.4, consiste de dois decodificadores concatenados serialmente através do *interleaver*. O processo de decodificação turbo utiliza o algoritmo BCJR. Neste algoritmo informações sobre a confiabilidade dos bits de um dos decodificadores (saída suave) , alimenta a entrada do outro decodificador (entrada suave), aumentando assim a confiabilidade de estimativa para cada bit.

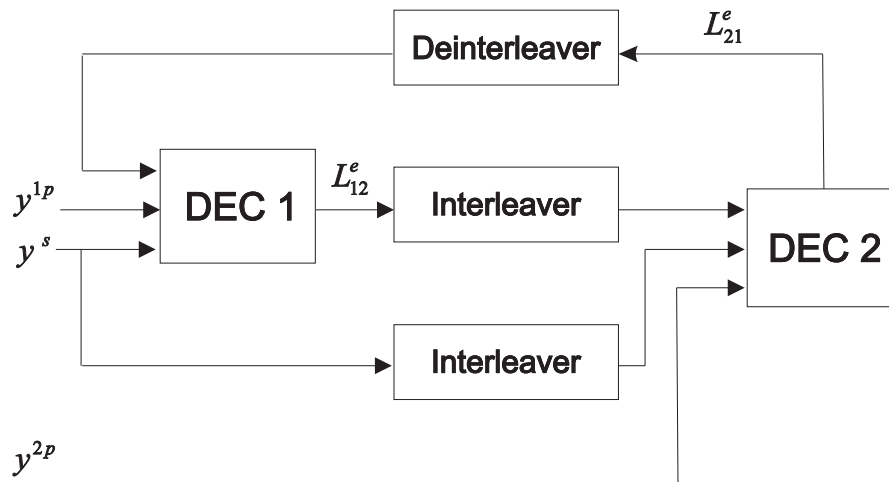


Figura 3.4: Diagrama de Blocos do Decodificador Turbo.

O primeiro decodificador (DEC 1) da Figura 3.4, tem como entrada a sequência  $y^s$  e a sequência de paridade  $y^{1p}$ , onde as sequências  $y^s$  e  $y^{1p}$  são as versões ruidosas de  $x^s$  e  $x^{p1}$ . O decodificador então produz uma saída suave, chamada de probabilidade extrínseca ( $L_{12}^e$ ), a qual é permutada pelo *interleaver* e é usada para melhorar a estimativa da probabilidade *a priori* da sequência de informação para o segundo decodificador.

As outras duas entradas do segundo decodificador (DEC 2) são a sequência  $y^s$  permutada e a sequência de paridade  $y^{2p}$ , onde a sequência  $y^{2p}$  é a versão ruidosa de  $x^{p2}$ . O segundo decodificador também produz uma saída suave, a probabilidade extrínseca  $L_{21}^e$ , a qual passa pelo *deinterleaver* e é usada para melhorar a estimativa da probabilidade *a priori* da sequência de informação para o primeiro decodificador.

Este processo continua iterativamente, e depois de um certo número de iterações é interrompido e o decodificador faz uma decisão abrupta para cada bit.

### 3.3.1 Algoritmo BCJR Modificado

O algoritmo de Viterbi é um método de decodificação por máxima verossimilhança que minimiza a probabilidade de erro de palavra para códigos convolucionais [30]. Mas este algoritmo não necessariamente minimiza a probabilidade de erro de bit. Em [7] os autores propuseram o algoritmo BCJR que minimiza a probabilidade de erro de bit.

Abaixo descrevemos o algoritmo BCJR modificado para o emprego na decodificação turbo, dado por [31]. Antes de começarmos a descrever o algoritmo algumas definições são necessárias:

- $m$  é a memória do codificador;
- $S$  é o conjunto de todos os  $2^m$  estados do codificador;
- $\mathbf{x}^s = (x_1^s, x_2^s, \dots, x_N^s) = (u_1, u_2, \dots, u_N)$  é a sequência de entrada do codificador;
- $\mathbf{x}^p = (x_1^p, x_2^p, \dots, x_N^p)$  é a paridade gerada por um codificador constituinte;
- $y_k = (y_k^s, y_k^p)$  é a saída do codificador  $(x_k^s, x_k^p)$  com ruído Gaussiano;
- $\mathbf{y}_a^b = (y_a, y_{a+1}, \dots, y_b)$
- $\mathbf{y} = \mathbf{y}_1^N = (y_1, y_2, \dots, y_N)$  é a palavra-código recebida com ruído;

### Decodificador MAP Iterativo

O decodificador MAP decide por  $u_k = +1$  se  $P(u_k = +1 | y) > P(u_k = -1 | y)$  e  $u_k = -1$  caso contrário. Mais sucintamente, a decisão  $u_k$  é dada por:

$$u_k = \text{sign}[L(u_k)] \quad (3.6)$$

onde  $L(u_k)$  é o logaritmo da relação da probabilidade *a posteriori* (LAPP) definida como:

$$L(u_k) = \log \left( \frac{P(u_k = +1 | y)}{P(u_k = -1 | y)} \right) \quad (3.7)$$

Associando os valores das transições de estado da treliça representada pela Figura 3.5, a equação (3.7) pode ser escrita como:

$$L(u_k) = \log \left( \frac{\sum_{S^+} p(s_{k-1} = s', s_k = s, y) | p(y)}{\sum_{S^-} p(s_{k-1} = s', s_k = s, y) | p(y)} \right) \quad (3.8)$$

onde  $s_k \in S$  é o estado do codificador no tempo  $k$ ,  $S^+$  é o par  $(s', s)$  correspondente a todas as transições de estado  $(s_{k-1} = s') \rightarrow (s_k = s)$  causado pelo dado de entrada  $u_k = +1$ , e similarmemente  $S^-$  para o dado de entrada  $u_k = -1$ . Na equação (3.8) observa-se que  $p(y)$  pode ser cancelado o que significa que precisamos somente calcular  $p(s_{k-1} = s', s_k = s, y)$ . A equação para este cálculo é dada em [7]:

$$p(s', s, y) = \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s) \quad (3.9)$$

onde  $\alpha_k(s) = p(s_k = s, y_1^k)$  é computado recursivamente como:

$$\alpha_k(s) = \sum_{s' \in S} \alpha_{k-1}(s') \gamma_k(s', s) \quad (3.10)$$

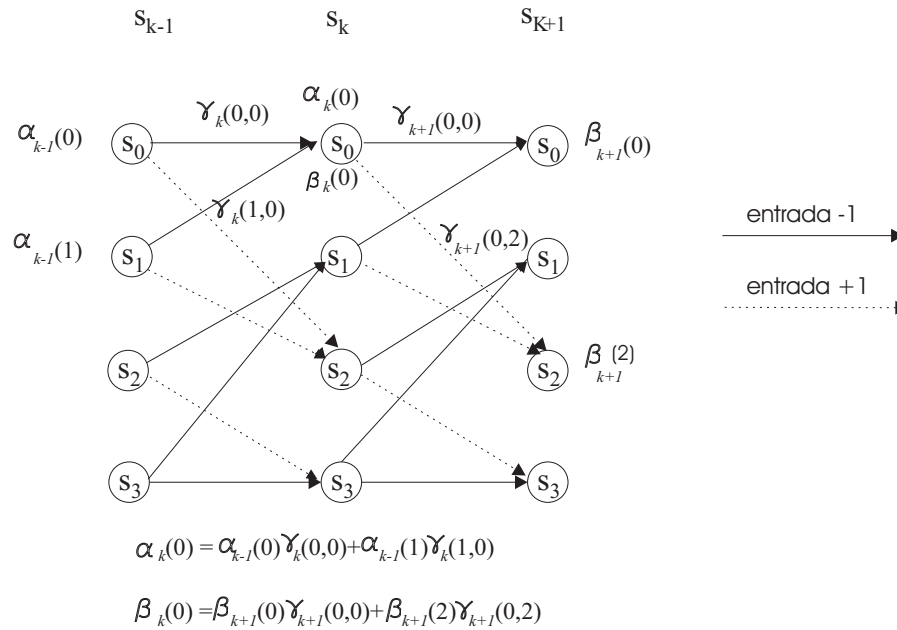


Figura 3.5: Cálculo de  $\alpha$  e  $\beta$ .

com condições iniciais  $\alpha_0(0) = 1$  e  $\alpha_0(s \neq 0) = 0$ , a probabilidade  $\gamma_k(s', s)$  na equação (3.10) será discutida com mais detalhes futuramente, e é definida como:

$$\gamma_k(s', s) = p(s_k = s, y_k | s_{k-1} = s') \quad (3.11)$$

A probabilidade  $\beta_k(s)$  na equação (3.9) é calculada recursivamente do final da treliça para o início como:

$$\beta_{k-1}(s') = \sum_{s \in S} \beta_k(s) \gamma_k(s', s) \quad (3.12)$$

com condições iniciais  $\beta_N(0) = 1$  e  $\beta_N(s \neq 0) = 0$ , "já que o codificador foi zerado".

A Figura 3.5 exemplifica o cálculo de  $\alpha$  e  $\beta$ , onde se deseja calcular os valores de  $\alpha_k$  e  $\beta_k$  no estado 0. A Figura apresenta as possíveis transições que terminam e partem do estado 0 e os respectivos cálculos de  $\alpha_k$  e  $\beta_k$  a partir das equações (3.10) e (3.12).

Porém, cancelar o termo  $p(y)$  na equação (3.8) resulta em um algoritmo instável [31]. Para resolver este problema podemos dividir  $p(y)$  por  $p(y_k)$ , deste modo obtemos:

$$\hat{\alpha}_k(s) = \alpha_k(s) / p(y_1^k) \quad (3.13)$$

e

$$\hat{\beta}_k(s) = \beta_k(s) / p(y_{k+1}^N | y_1^k) \quad (3.14)$$

A equação (3.9) se modifica para:

$$p(s', s, y)p(y_k) = \hat{\alpha}_{k-1}(s')\gamma_k(s', s)\hat{\beta}_k(s) \quad (3.15)$$

Note que desde  $p(y_1^k) = \sum_{s \in S} \alpha_k(s)$ , os valores  $\hat{\alpha}_k(s)$  podem ser calculados por:

$$\hat{\alpha}_k(s) = \alpha_k(s) / \sum_{s \in S} \alpha_k(s) \quad (3.16)$$

Podemos usar a equação (3.10) na equação (3.16) para obter a recursão envolvendo unicamente  $\hat{\alpha}_k(s)$

$$\hat{\alpha}_k(s) = \frac{\sum_{s'} \alpha_{k-1}(s')\gamma_k(s', s)}{\sum_s \sum_{s'} \alpha_{k-1}(s')\gamma_k(s', s)} = \frac{\sum_{s'} \hat{\alpha}_{k-1}(s')\gamma_k(s', s)}{\sum_s \sum_{s'} \hat{\alpha}_{k-1}(s')\gamma_k(s', s)} \quad (3.17)$$

onde a segunda igualdade é obtida dividindo o numerador e denominador por  $p(y_1^{k-1})$ . A recursão para  $\hat{\beta}_k(s)$  pode ser obtida como:

$$\begin{aligned} p(y_k^N | y_1^{k-1}) &= p(y_1^k) \frac{p(y_{k+1}^N | y_1^k)}{p(y_1^{k-1})} \\ &= \sum_s \sum_{s'} \alpha_{k-1}(s')\gamma_k(s', s) \frac{p(y_{k+1}^N | y_1^k)}{p(y_1^{k-1})} \\ &= \sum_s \sum_{s'} \hat{\alpha}_{k-1}(s')\gamma_k(s', s)p(y_{k+1}^N | y_1^k) \end{aligned} \quad (3.18)$$

dividindo (3.12) por esta equação obtém-se:

$$\hat{\beta}_{k-1}(s') = \frac{\sum_s \hat{\beta}_k(s)\gamma_k(s', s)}{\sum_s \sum_{s'} \hat{\alpha}_{k-1}(s')\gamma_k(s', s)} \quad (3.19)$$

Combinando a equação (3.8) e a equação (3.15) obtemos a LAPP para cada bit:

$$L(u_k) = \log \left( \frac{\sum_{S^+} \hat{\alpha}_{k-1}(s')\gamma_k(s', s)\hat{\beta}_k(s)}{\sum_{S^-} \hat{\alpha}_{k-1}(s')\gamma_k(s', s)\hat{\beta}_k(s)} \right) \quad (3.20)$$

### Cálculo da informação extrínseca

Pela regra de Bayes a LAPP para cada bit pode ser escrita como:

$$L(u_k) = \log \left( \frac{P(y|u_k = +1)}{P(y|u_k = -1)} \right) + \log \left( \frac{P(u_k = +1)}{P(u_k = -1)} \right)$$

onde o segundo termo representa a informação *a priori*. Se  $P(u_k = +1) = P(u_k = -1)$ , a informação *a priori* é nula. Durante o processo iterativo, o primeiro decodificador produzirá uma

informação extrínseca, que será considerada pelo segundo decodificador como a nova informação *a priori*. Similarmente o segundo decodificador produzirá uma informação extrínseca que será considerada pelo primeiro decodificador como a nova informação *a priori* e assim por diante.

Abaixo mostraremos como a informação extrínseca é obtida do algoritmo BCRJ modificado [31]. A equação (3.11) pode ser escrita como:

$$\gamma_k(s', s) = P(s|s')p(y_k|s', s) = P(u_k)p(y_k|u_k)$$

onde  $u_k$  corresponde a transição do estado  $s' \rightarrow s$ . Definindo

$$L^e(u_k) = \log \left( \frac{P(u_k = +1)}{P(u_k = -1)} \right)$$

podemos escrever  $P(u_k)$  como sendo:

$$P(u_k) = \left( \frac{\exp[-L^e(u_k)/2]}{1 + \exp[-L^e(u_k)/2]} \right) \exp[u_k L^e(u_k)/2] \quad (3.21)$$

Esta igualdade é obtida através das seguintes considerações:

$$P(u_k = +1) = P^+ = \sqrt{P^+/P^-} \left( \frac{\sqrt{P^-/P^+}}{1 + P^-/P^+} \right)$$

$$P(u_k = -1) = P^- = \sqrt{P^-/P^+} \left( \frac{\sqrt{P^-/P^+}}{1 + P^-/P^+} \right)$$

$P(y_k|u_k)$  é dado por:

$$\begin{aligned} P(y_k|u_k) &= \exp \left[ -\frac{(y_k^s - u_k)^2}{2\sigma^2} - \frac{(y_k^p - x_k^p)^2}{\sigma^2} \right] \\ &= \exp \left[ \frac{y_k^{s2} + u_k^2 + y_k^{p2} + x_k^{p2}}{\sigma^2} \right] \exp \left[ \frac{y_k^s u_k + y_k^p x_k^p}{\sigma^2} \right] \end{aligned} \quad (3.22)$$

Então, com algumas simplificações, obtemos a equação para o cálculo de  $\gamma_k(s', s)$ :

$$\gamma_k(s', s) = \exp[u_k L^e(u_k)/2] \exp \left[ \frac{u_k y_k^s + x_k^p y_k^p}{\sigma^2} \right] \quad (3.23)$$

Esta equação pode ser reescrita como:

$$\gamma_k(s', s) = \exp \left[ \frac{1}{2} u_k (L^e(u_k) + L_c y_k^s) \right] \exp \left[ \frac{1}{2} L_c y_k^p x_k^p \right] \quad (3.24)$$



onde  $L_c = \frac{4E_b}{N_0}$  sendo  $E_b$  a energia por bit. Combinando a equação (3.24) e a equação (3.20) obtemos:

$$\begin{aligned} L(u_k) &= \log \left( \frac{\sum_{S^+} \hat{\alpha}_{k-1}(s') \gamma_k^e(s', s) \hat{\beta}_k(s) C_k}{\sum_{S^-} \hat{\alpha}_{k-1}(s') \gamma_k^e(s', s) \hat{\beta}_k(s) C_k} \right) \\ &= L_c y_k^s + L^e(u_k) + \log \left( \frac{\sum_{S^+} \hat{\alpha}_{k-1}(s') \gamma_k^e(s', s) \hat{\beta}_k(s)}{\sum_{S^-} \hat{\alpha}_{k-1}(s') \gamma_k^e(s', s) \hat{\beta}_k(s)} \right) \end{aligned} \quad (3.25)$$

onde  $C_k = \exp \left[ \frac{1}{2} u_k (L^e(u_k) + L_c y_k^s) \right]$ .

O primeiro termo da equação 3.25 é chamado algumas vezes de *valor do canal*, o segundo termo representa a informação *a priori* e o terceiro termo representa a informação extrínseca. Assim por exemplo, para uma dada iteração o primeiro decodificador calcula:

$$L_1(u_k) = L_c y_k^s + L_{21}^e(u_k) + L_{12}^e(u_k)$$

onde  $L_{21}^e(u_k)$  é a informação extrínseca passada do segundo decodificador para o primeiro decodificador e  $L_{12}^e(u_k)$  é a informação extrínseca do primeiro decodificador que será passada para o segundo decodificador.

## 3.4 Desempenho do Código Turbo

Nesta seção foram feitas simulações de um código turbo para avaliar o seu desempenho para o canal AWGN com modulação BPSK. As simulações deste capítulo foram feitas em linguagem C baseadas nos códigos dados em [35]. Foi obtida a curva com a eficiência espectral pela relação  $E_b/N_0$  para a modulação BPSK. O desempenho do código foi comparado variando o número de iterações e o tamanho do *interleaver*, e por último seu desempenho foi comparado com um código convolucional decodificado utilizando o algoritmo de Viterbi.

### 3.4.1 Capacidade de Canal

A curva da Figura 3.6 mostra a eficiência espectral pela relação  $E_b/N_0$  para a modulação BPSK, com probabilidade de erro de  $10^{-5}$ . A modulação BPSK alcança esta taxa de erro em  $E_b/N_0 = 9.5$  dB e eficiência espectral de 1 bits/s/Hz, enquanto que o limite de Shannon para este caso é de 0 dB. Assim este esquema é em torno de 9.5 dB pior do que o limite de Shannon.

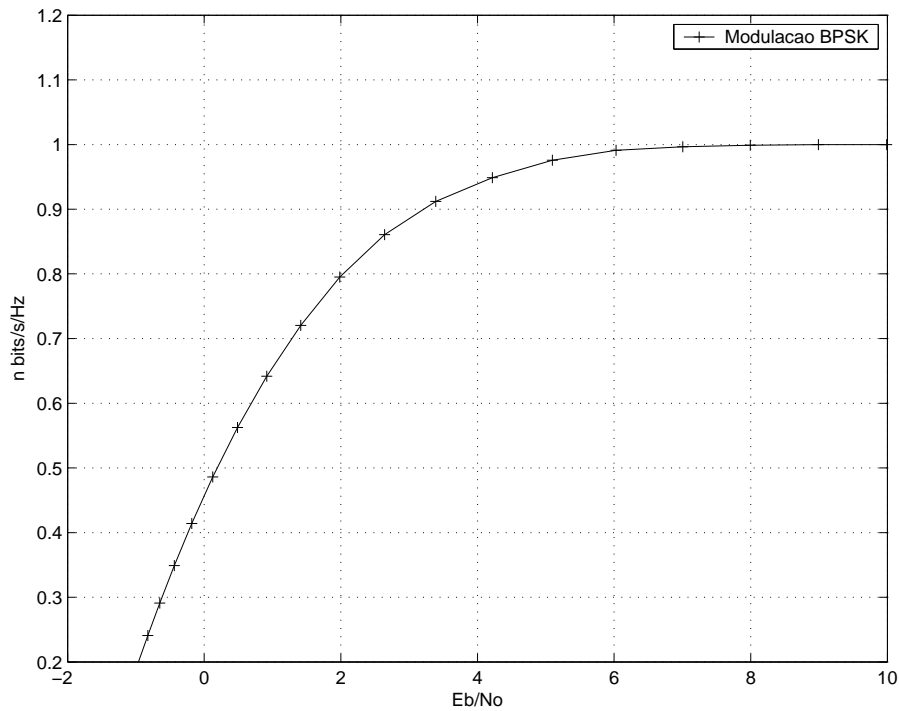


Figura 3.6: Eficiência Espectral por  $E_b/N_0$  para a Modulação BPSK.

A curva também mostra a máxima eficiência espectral que o sistema pode operar para uma dada relação  $E_b/N_0$ , por exemplo se  $E_b/N_0 = 0$  dB a máxima eficiência espectral alcançada é 0.5 bit/s/Hz podendo ser utilizada a modulação BPSK sem a necessidade de uma modulação com ordem e complexidade maior nesta faixa. Por este motivo, como neste trabalho estamos interessados na região de baixa relação sinal ruído consideraremos apenas modulação BPSK.

### 3.4.2 Efeito do Número de Iterações no Desempenho

Nesta simulação foi utilizado um código turbo de taxa 1/3, com polinômios geradores  $g_0 = 37$  e  $g_1 = 21$ , na forma octal,  $d_{free} = 6$ , comprimento de restrição  $v = 5$ , e *interleaver* com tamanho  $N = 1024$  bits. O resultado da simulação, variando o número de iterações, está mostrado na Figura 3.7.

Ao aumentar o número de iterações de uma para duas iterações ocorreu uma significativa melhora no desempenho do código, de duas iterações para quatro também há uma melhora significativa no desempenho. De quatro para sete iterações a melhora já não é mais tão significativa, há uma saturação no desempenho do decodificador. Isto significa que para este código mais do que sete iterações resultaria em pouca melhora no desempenho do decodificador e a complexidade de decodificação seria maior.

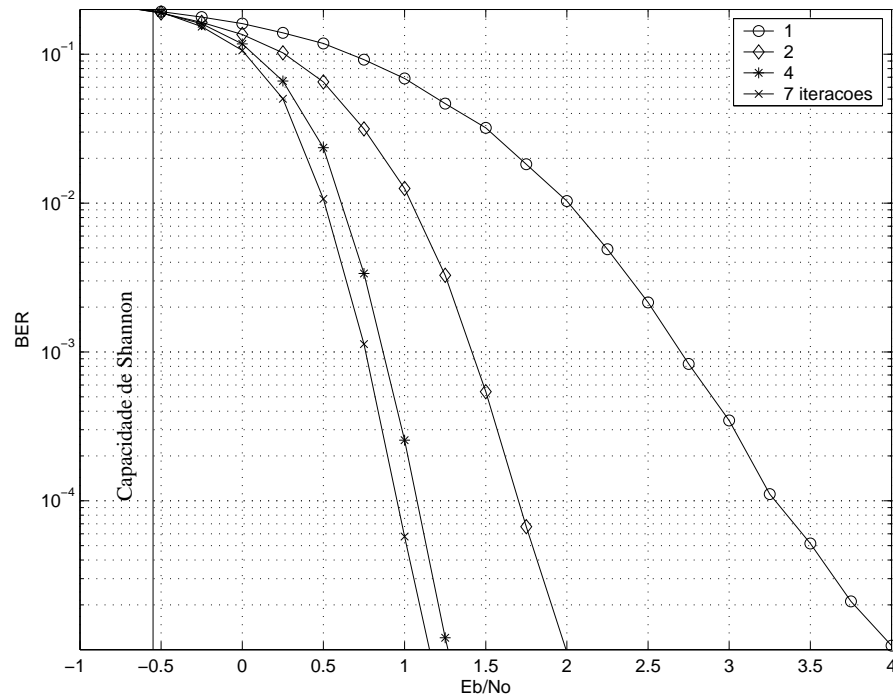


Figura 3.7: Taxa de erro de Bit no canal AWGN com diferentes números de iterações simulado com codificador turbo de taxa 1/3 e  $L=1024$ .

Para evitar iterações desnecessárias e atraso de decodificação quando o decodificador se aproxima de seu limite de desempenho, é importante um critério eficiente de parada. Um método que pode ser usado é o *cyclic redundancy check* (CRC). A informação de entrada é codificada pelo CRC e depois disso é codificada pelo codificador turbo. Se na saída do decodificador o CRC não detectar erro o processo iterativo é terminado. Um outro critério de parada que pode ser utilizado é a média da confiabilidade de cada bit. Esta média é definida como a média dos valores absolutos do logaritmo da relação da probabilidade de cada bit (LLR) [32]. Depois de um certo número de iterações a média de confiabilidade de cada bit não muda significativamente. Assim, quando a média de confiabilidade dos dois decodificadores não mudar em duas iterações subsequentes, o processo de decodificação pode ser interrompido

### 3.4.3 Efeito do Tamanho do Interleaver no Desempenho

Para comparar o efeito do *interleaver* no desempenho do código turbo consideramos o mesmo número de iterações e tamanhos de *interleavers* diferentes. O sistema foi simulado usando taxa 1/3, com polinômios geradores  $g_0 = 37$  e  $g_1 = 21$ , oito iterações e *interleavers* de tamanhos 100, 512, 1024, 2048 e 4098.

Claramente podemos notar pela Figura 3.8 que o desempenho melhora a medida que au-

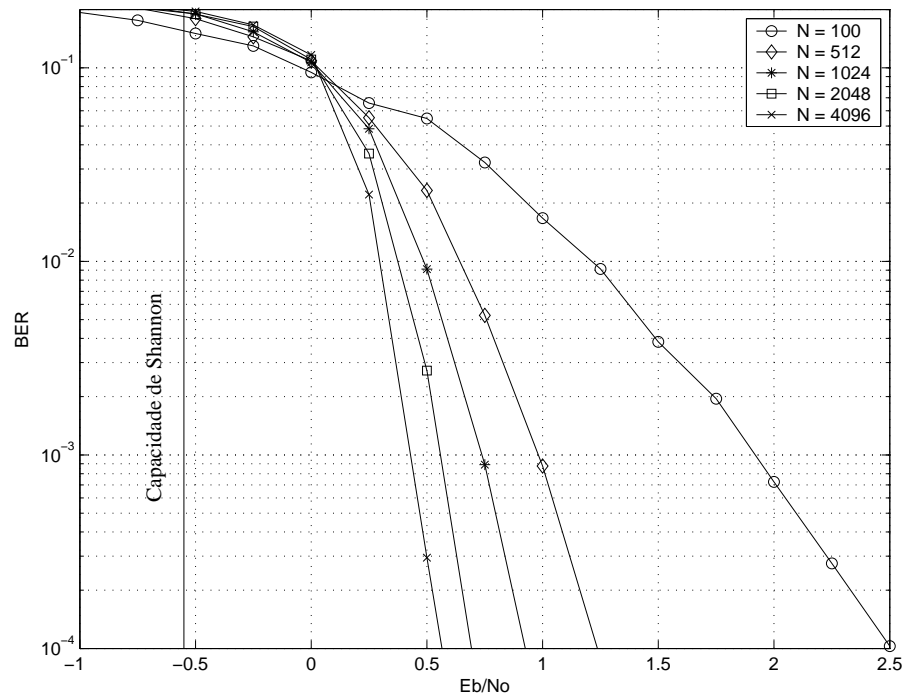


Figura 3.8: Taxa de erro de Bit (BER) pela relação  $E_b/N_0$  para o codificador turbo de taxa 1/3 e oito iterações.

mentamos o tamanho do *interleaver*. O *interleaver* de tamanho  $N=2048$  tem desempenho melhor do que o *interleaver* de tamanho  $N=1024$  bits,  $N=512$  bits e  $N=100$  bits. E o *interleaver* de tamanho  $N=4098$  bits apresenta desempenho melhor do que o *interleaver* de  $N=2048$  bits, ficando distance 1.1dB da capacidade de Shannon.

Aumentar o tamanho do *interleaver* tem o efeito de randomizar a sequência de informação na entrada do segundo decodificador. Conseqüentemente, as entradas dos dois decodificadores se tornam menos correlacionadas em relação ao ruído, melhorando a performance da decodificação.

### 3.4.4 Desempenho do Código Convolutacional Versus desempenho do Código Turbo

A Figura 3.9 compara o desempenho do código turbo com taxa 1/3, polinômio gerador  $g_0 = 37$  e  $g_1 = 21$ , oito iterações, e *interleaver* com  $N=4096$  bits, com o código convolutacional com taxa de codificação 1/3 e polinômio gerador  $(25,33,37)$ ,  $v = 5$ ,  $d_{free} = 12$ , e decodificado utilizando o algoritmo de Viterbi com decisão abrupta.

O código turbo apresenta uma taxa de erro de bit bem menor do que o código convolutacional. Para a taxa de erro de bit de  $10^{-4}$ , o código turbo fica em torno de apenas 1.1 dB da capacidade

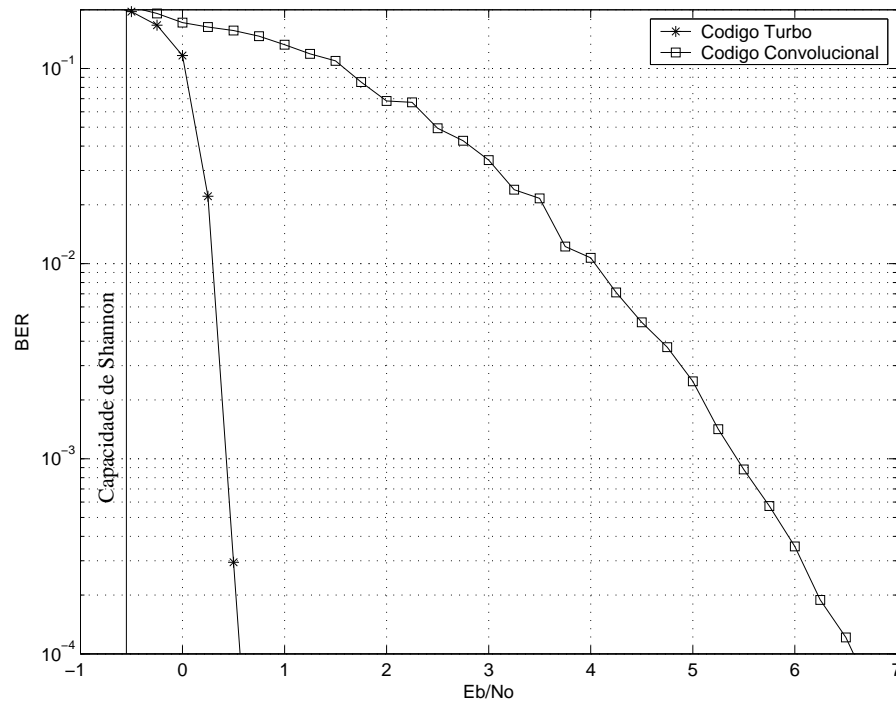


Figura 3.9: Taxa de erro de bit (BER) por  $E_b/N_0$  para o código Turbo e o Código Convolucional de taxa 1/3.

de Shannon, enquanto que o código convolucional fica distante em torno de 7dB da capacidade de Shannon.

### 3.5 Comentários Finais

Neste capítulo foram apresentados com mais detalhes os códigos turbo. Foi apresentada a estrutura do codificador e do decodificador e o algoritmo utilizado para a decodificação, o algoritmo MAP. Foram feitas simulações no computador para avaliar o seu desempenho em relação ao número de iterações, tamanho do interleaver e por último uma comparação do seu desempenho com o código convolucional decodificado pelo algoritmo de Viterbi.

## 4 *Esquema de ARQ Híbrido Baseado nos Códigos Convolucionais Recursivos e na Decodificação Turbo*

### 4.1 Introdução

A introdução dos códigos turbo [33] incitou muita pesquisa no campo da decodificação iterativa. A codificação turbo é uma técnica de correção de erro muito eficiente, alcançando um desempenho consideravelmente perto da capacidade do canal. Porém, embora os códigos turbo apresentem um bom desempenho em relação à taxa erro de bit, eles não garantem uma transmissão de dados livre de erros. Quando uma transmissão de dados livre de erros for um requisito do sistema (por exemplo, uma transferência bancária), o protocolo de controle de erro (ARQ) [18] deve ser utilizado.

Nos últimos anos, mecanismos de ARQ em conjunto com códigos turbo foram consideravelmente investigados. Um dos primeiros trabalhos no assunto foi feito por Narayanan e Stüber [34]. Os autores propuseram um esquema ARQ/FEC híbrido onde a mensagem de informação é codificada pelo codificador turbo antes da transmissão. Na primeira transmissão, o receptor opera como um decodificador turbo regular onde nenhuma informação *a priori* sobre a mensagem de informação é conhecida (igual às probabilidades *a priori* de '0's e '1's). Se for detectado erro no receptor, então um pedido de retransmissão é enviado ao transmissor. O transmissor então envia os bits codificados novamente, mas na segunda transmissão o receptor opera como um decodificador turbo modificado. A decisão suave da última transmissão é utilizada como a nova informação *a priori*. Com esta simples e engenhosa modificação, como foi mostrado em [34], uma melhora de desempenho considerável pode ser alcançada.

Uma técnica análoga foi proposta em [36], onde os bits de paridade são incrementalmente transmitidos de acordo com as condições do canal. Neste esquema, na primeira transmissão somente os bits sistemáticos são enviados. Se o transmissor receber um pedido para retransmis-

são, então ele começa a transmissão dos bits de paridade, os quais são enviados somente quando necessário. Em [37], um esquema de redundância incremental baseado na decodificação turbo também foi utilizado. Inicialmente, são transmitidos só os bits sistemáticos e alguns bits de paridade. Se um pedido de retransmissão é recebido no transmissor, então são enviados bits de paridade adicionais, de acordo com diferentes condições do canal. O procedimento é repetido até que todos os bits de paridade sejam transmitidos. Em [38] os autores propuseram um esquema baseado na confiabilidade dos pacotes recebidos. A confiabilidade de pacote recebido foi obtida calculando o valor médio da *LLR* dos bits dentro de cada pacote. A informação de confiabilidade das transmissões anteriores é acumulada no receptor e é usada para determinar o tamanho dos pacotes das transmissões subseqüentes.

Em [39] diferentes técnicas de combinação de pacotes baseadas na *LLR* de cada bit recebido foram apresentadas. Foram propostos três métodos. O melhor desempenho foi obtido pelo esquema que utilizou uma técnica semelhante à proposta em [34]. Em [40] os autores propuseram um método baseado na observação que a concatenação paralela dos códigos convolucionais (PCCC) pode ser representada por uma equivalente concatenação serial dos códigos convolucionais (SCCC) puncionados. Na primeira transmissão o esquema utiliza um PCCC. Se um pedido de retransmissão é feito, então o sistema transmite bits de paridade adicionais tais que, em conjunto com os bits previamente transmitidos, constituem um código SCCC equivalente.

Neste capítulo nós propomos um novo esquema ARQ híbrido baseado nos códigos convolucionais recursivos e na decodificação turbo. O método é comparado com o proposto por Narayanan e Stüber [34], tanto em termos de desempenho da taxa de erro como da complexidade de decodificação. Nossos resultados mostram que, em geral, o esquema proposto alcança uma menor taxa de erro de frame (FER) com uma complexidade de decodificação menor. Além disso, ambos os métodos gastam a mesma energia para transmitir um determinado número de bits de informação.

## 4.2 Modelo do Sistema

Considere um esquema ARQ/FEC híbrido para a transmissão de um bloco de mensagem de  $L$  bits equiprováveis e independentes,  $u = [u_0 u_1 \cdots u_{L-1}]$ , através de um canal AWGN com ruído de densidade espectral  $\frac{N_0}{2}$ . Sem perda de generalidade, o esquema de ARQ que empregamos é o protocolo *stop-and-wait* [18]. Supomos ainda que erros podem ser detectados perfeitamente no receptor. Na prática esta capacidade pode ser implementada por um codificador e decodificador *cyclic redundancy check* (CRC). Quando for detectado erro no receptor, um pedido de

retransmissão (NACK) é enviado ao transmissor. Quando o pacote é recebido corretamente, um sinal de reconhecimento (ACK) é enviado ao transmissor o qual transmite o próximo pacote. Assumimos que o canal de retorno (entre o receptor e transmissor) é livre de erros. O diagrama de blocos com os elementos principais deste sistema de comunicações está mostrado na Figura 4.1.

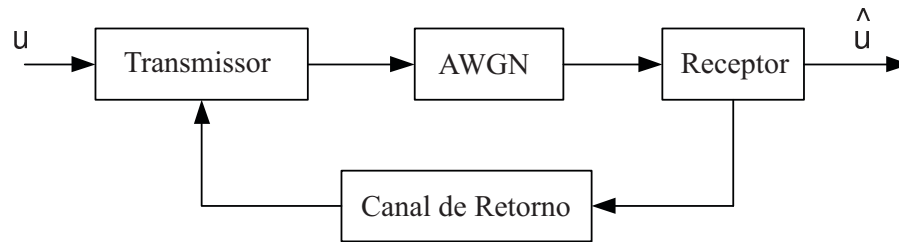


Figura 4.1: Diagrama de blocos de um sistema de comunicação empregando o protocolo ARQ.

Antes da primeira transmissão, cada bloco de  $L$  bits de mensagem  $u$  é codificado por um codificador convolucional sistemático recursivo (RCC) de taxa  $1/2$ , resultando em uma sequência de bits sistemáticos e de paridade  $v = [v^s; v^{p1}]$  de comprimento  $2L$ , onde  $v^s = [v_0^s v_1^s \cdots v_{L-1}^s]$  e  $v^{p1} = [v_0^{p1} v_1^{p1} \cdots v_{L-1}^{p1}]$ . Esta sequência é então mapeada para BPSK, resultando em um bloco de  $2L$  símbolos BPSK  $x = [x^s; x^{p1}]$ ,  $x^s = [x_0^s x_1^s \cdots x_{L-1}^s]$  e  $x^{p1} = [x_0^{p1} x_1^{p1} \cdots x_{L-1}^{p1}]$ . Estes símbolos serão transmitidos pelo canal usando  $E = 2L \times E_s$  unidades de energia, onde  $E_s$  é a energia por símbolo. Com as definições acima os  $2L$  símbolos no receptor,  $y = [y^s; y^{p1}]$ , podem ser escritos como <sup>1</sup>:

$$y = x + w = (x^s + w^s) + (x^{p1} + w^{p1}) \quad (4.1)$$

onde  $w = [w^s; w^{p1}]$  é um vetor com comprimento  $2L$  é uma variável aleatória Gaussiana com média zero e variância  $\sigma^2 = \frac{N_0}{2}$ . Baseado nos símbolos recebidos  $y$  o receptor produz uma estimativa  $\hat{u}$  dos bits de mensagem. Esta estimativa é obtida por meio do algoritmo BCJR [7], uma vez que as saídas suaves do decodificador podem ser necessárias para a decodificação das mensagens retransmitidas. Depois da decodificação, o receptor confere pelo CRC se a mensagem recebida foi corrompida ou não. No caso do receptor detectar erro, uma mensagem de NACK é enviada para transmissor o qual procederá com a primeira retransmissão.

<sup>1</sup>O esquema pode ser generalizado facilmente para outras taxas de codificação e para outras ordens de modulação.



### 4.2.1 Segunda Transmissão

Se o transmissor receber um NACK então para a segunda transmissão (primeira retransmissão) o bloco de  $L$  bits de mensagem  $u$  é primeiramente permutado pelo *interleaver* e então codificado por um RCC de taxa 1, resultando em uma sequência  $v^{p2}$  de bits de paridade, de comprimento  $L$ . Esta sequência é mapeada para a modulação BPSK resultando em um bloco de  $L$  símbolos BPSK,  $x^{p2}$ . Para manter a mesma energia gasta para cada transmissão, cada símbolo da sequência  $x^{p2}$  é transmitido com energia  $2E_s$ . Os símbolos recebidos são definidos como  $y^{p2} = x^{p2} + w^{p2}$ , onde  $w^{p2}$  é outra sequência de ruído gaussiano com média zero e variância  $\sigma^2$ . O receptor procede então como se a decodificação da primeira transmissão fosse a primeira metade de uma iteração de um decodificador turbo [33]. A saída suave da decodificação da primeira transmissão é usada como informação *a priori* para o segundo decodificador. Além disso, os símbolos recebidos depois da primeira transmissão, os quais correspondem aos bits sistemáticos,  $y^s$ , são permutados e utilizados em conjunto com  $y^{p2}$  para produzir a saída suave do segundo decodificador. Então, como um decodificador turbo regular, esta informação suave alimenta o primeiro decodificador como a informação *a priori* que utiliza  $y^s$  e  $y^{p1}$  para produzir uma nova saída suave. O decodificador opera um determinado número de iterações e produz uma estimativa  $\hat{u}$  dos bits de mensagem.

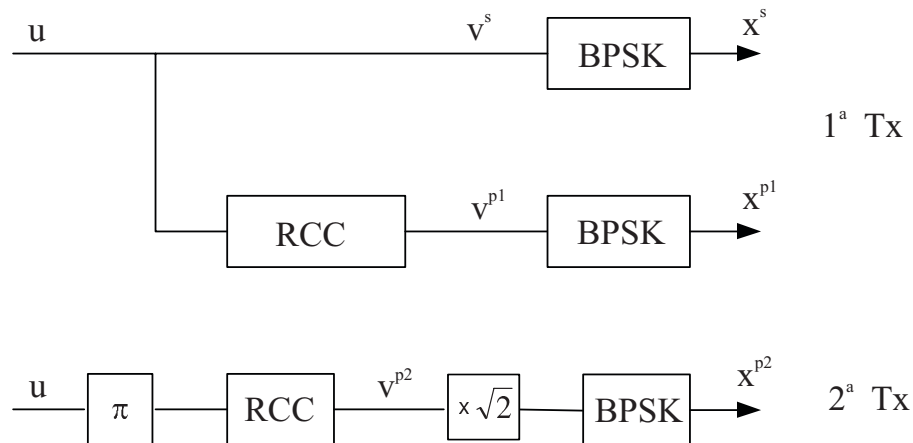


Figura 4.2: Diagrama de blocos com a estratégia de transmissão do esquema proposto, considerando a primeira e a segunda transmissões. RCC significa um codificador convolucional recursivo, enquanto que  $\pi$  é um *interleaver* de tamanho  $L$ .

A Figura 4.2 mostra o diagrama de blocos para a estratégia de transmissão para a primeira e segunda transmissões. Se o receptor ainda detectar erro na mensagem recebida, outra mensagem de NACK é enviada ao transmissor o qual procederá com a terceira transmissão. Porém, na terceira transmissão não é necessário codificar os dados novamente, uma vez que os dados

codificados ( $v^s$ ,  $v^{p1}$  e  $v^{p2}$ ) podem ser armazenados no transmissor até que uma mensagem de ACK seja recebida.

Número de Tx's	Estratégia de Codificação	Estratégia de Transmissão	Energia por Símbolo	Estratégia de Decodificação
1ª Tx	Codifica $u$ com RCC sistemático.	Envia $x^s$ e $x^{p1}$	$E_s$	Decodifica $y^s$ e $y^{p1}$ como decodificador convolucional
2ª Tx	Permuta $u$ e codifica com RCC.	Envia $x^{p2}$	$2E_s$	Decodifica $y^s$ , $y^{p1}$ e $y^{p2}$ como decodificador turbo
3ª Tx	-	Envia $x^s$	$2E_s$	Decodifica $y^s$ , $y^{p1}$ e $y^{p2}$ como decodificador turbo
4ª Tx	-	Envia $x^{p1}$	$2E_s$	Decodifica $y^s$ , $y^{p1}$ e $y^{p2}$ como decodificador turbo
5ª Tx	-	Envia $x^{p2}$	$2E_s$	Decodifica $y^s$ , $y^{p1}$ e $y^{p2}$ como decodificador turbo
6ª Tx em diante	Repete da 3ª Tx em diante			

Tabela 4.1: Resumo das estratégias de codificação, de transmissão e decodificação para o esquema proposto.

## 4.2.2 Terceira Transmissão

No caso de uma terceira transmissão (segunda retransmissão) ser necessária, a parte sistemática  $v^s$  da mensagem codificada é transmitida novamente (sequência  $x^s$ ). Porém, para manter a mesma energia gasta em cada transmissão, a energia dos símbolos  $x^s$  é aumentada para  $2E_s$ . Apesar de retransmitirmos somente  $L$  símbolos, a energia total gasta na terceira transmissão é de  $2L \times E_s$ , a qual é igual à da primeira e segunda transmissões.

O receptor então substitui o vetor dos símbolos recebidos  $y^s$ , os quais chegaram depois da primeira transmissão, pelo novo vetor de símbolos recebidos  $y^s$ , os quais foram recebidos depois da terceira transmissão. Com esta substituição, e utilizando a saída suave produzida depois da segunda transmissão e os vetores  $y^{p1}$  e  $y^{p2}$ , o receptor executa a decodificação turbo dos símbolos recebidos e tenta estimar os bits de mensagem novamente. Se depois da decodificação turbo ainda for detectado erro, o transmissor procede com a quarta transmissão.

## 4.2.3 Quarta transmissão

Se necessário, na quarta transmissão (terceira retransmissão) os bits de paridade de saída do primeiro codificador constituinte,  $v^{p1}$ , são transmitidos. Novamente, desde que só  $L$  símbolos

serão transmitidos, os símbolos modulados  $x^{p1}$  tem sua energia aumentada para  $2E_s$ . O receptor substitui o vetor recebido anterior  $y^{p1}$  pelo novo vetor recebido  $y^{p1}$  depois da quarta transmissão. A decodificação continua similarmente ao caso da terceira transmissão.

#### 4.2.4 Transmissões Subseqüentes

Se for necessário uma quinta transmissão, sua estrutura é definida exatamente como para a segunda transmissão. A sexta transmissão da mesma maneira que a terceira transmissão, enquanto que uma sétima transmissão operaria como uma quarta, e assim por diante. A Tabela 4.1 resume as características da transmissão e estratégia de decodificação do esquema proposto.

### 4.3 Resultados de Simulação

Nesta seção investigamos o desempenho do esquema proposto em termos da taxa de erro de bloco ( $FER$ ) e de seu complemento  $\Gamma = 1 - FER$ , o qual pode ser mostrado proporcional ao *throughput* [22]. Para todas as simulações consideramos um codificador convolucional recursivo com polinômios geradores  $(17, 15)_8$  na forma octal,  $v = 4$  e  $d_{free} = 6$ . O bloco de mensagem tem comprimento de  $L = 97$  bits, e no caso da decodificação turbo foram feitas cinco iterações. Consideramos o esquema proposto por Narayanan e Stüber em [34] como *benchmark* para avaliar o desempenho do método proposto.

A Figura 4.3 mostra a  $FER$  por  $E_b/N_0$ , onde  $E_b$  é a energia por bit de informação, para a primeira e segunda transmissão do esquema proposto e do método em [34]. Para a primeira transmissão a  $FER$  do esquema proposto tem um desempenho pior do que o esquema em [34], mas para a segunda transmissão o esquema proposto já supera o método em [34]. Isto é devido ao fato de que para a primeira transmissão o esquema proposto opera como um código convolucional de taxa  $1/2$ , enquanto que na segunda transmissão o esquema proposto opera como um decodificador turbo de taxa  $1/3$ . Além disso, desde que um menor número de símbolos são transmitidos durante as retransmissões no esquema proposto (metade do número de símbolos que são transmitidos em cada retransmissão do método em [34]), o dobro de energia pode ser alocada para estes símbolos. Portanto, quando mais do que uma transmissão é necessária (ou quando NACK é enviada para o transmissor) o esquema proposto tem um melhor desempenho da taxa de erro do que o método em [34].

A Figura 4.4 mostra  $\Gamma = 1 - FER$  por  $E_b/N_0$  depois de duas transmissões para o esquema proposto, o método em [34], e para o caso do esquema ARQ/FEC híbrido regular usando código

turbo (neste caso, a diferença para o método em [34] é que a informação suave da transmissão anterior não é utilizada para a decodificação de uma retransmissão). Pela figura podemos ver que o método proposto em [34] alcança uma considerável melhora sobre o esquema FEC/ARQ híbrido regular usando o código turbo, e que o esquema proposto supera ambos os métodos.

Na Figura 4.5 mostramos  $\Gamma = 1 - FER$  para o esquema proposto depois da segunda e terceira transmissões. Para a terceira transmissão consideramos a retransmissão dos símbolos sistemáticos  $x^s$  ou os símbolos de paridade  $x^{p1}$ . Pela figura podemos ver que um melhor desempenho é obtido quando os bits sistemáticos são transmitidos novamente na terceira transmissão, ao invés de transmitir os símbolos  $x^{p1}$ . Esta é a razão pela qual na terceira transmissão (Seção 4.2.2) nós consideramos a retransmissão dos bits sistemáticos.

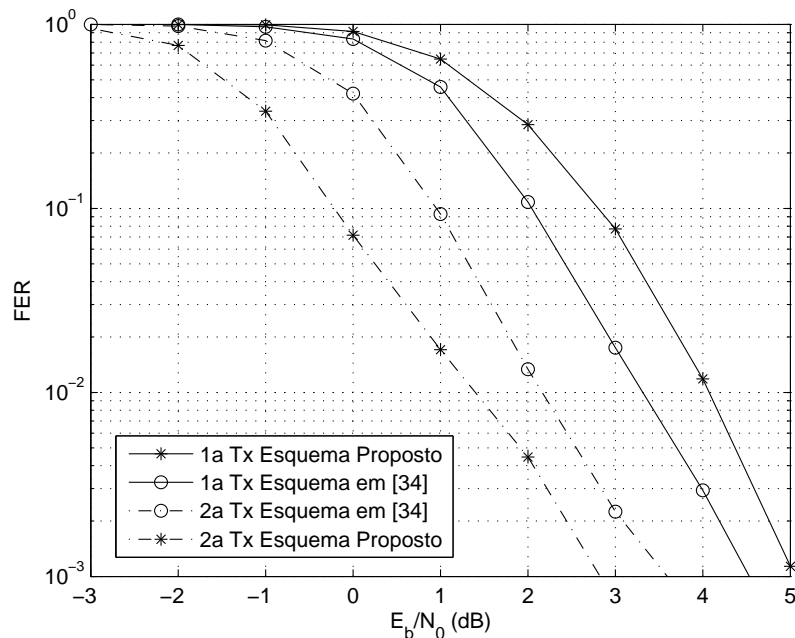


Figura 4.3: FER por  $E_b/N_0$  para a primeira e segunda transmissão considerando o esquema proposto e o método em [34].

Na próxima seção nós comparamos o esquema proposto neste trabalho com o esquema proposto em [34] em relação à complexidade computacional no receptor.

## 4.4 Análise da Complexidade Computacional

McEliece e Lin definiram em [41] a complexidade da treliça do código convolucional como o número de símbolos, por bit codificado, para cada módulo da treliça. Esta medida é diretamente relacionada ao esforço computacional necessário pelo algoritmo de Viterbi ou BCJR

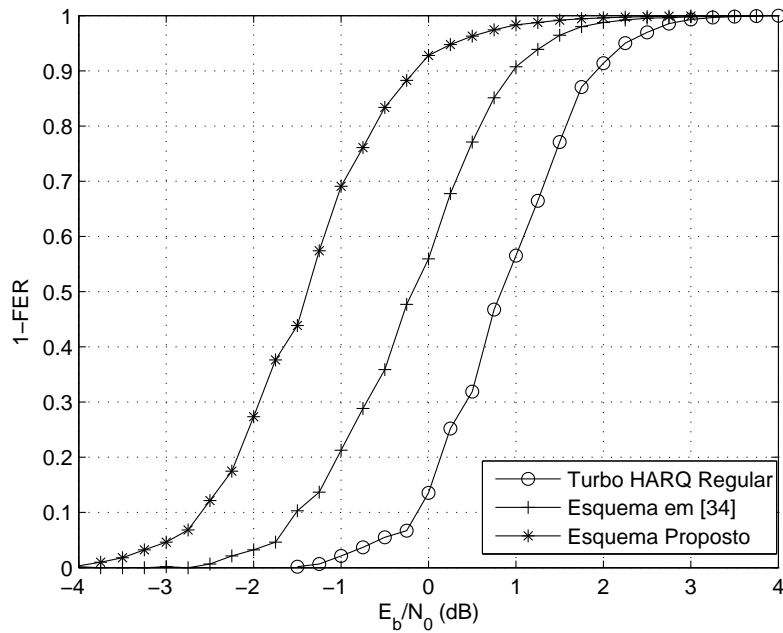


Figura 4.4:  $\Gamma = 1 - FER$  por  $E_b/N_0$  depois da segunda transmissão para o esquema ARQ/FEC híbrido regular usando código turbo, o esquema ARQ/FEC híbrido modificado usando o código turbo em [34], e o método proposto neste capítulo.

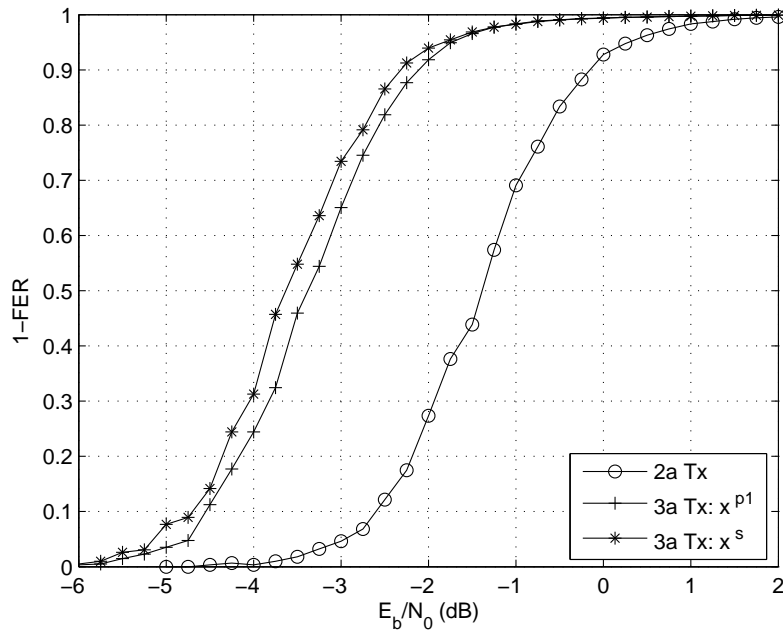


Figura 4.5:  $\Gamma = 1 - FER$  depois da terceira transmissão para o esquema proposto. Duas estratégias são consideradas: i) a transmissão dos símbolos de paridade  $x^{p1}$ ; ii) a transmissão dos símbolos sistemáticos  $x^s$ . Na curva também é mostrado o desempenho depois da segunda transmissão.

para decodificar um bit. Sejam  $n$ ,  $k$  e  $m$  definidos como o número de bits codificados, o número de bits de informação e a memória, respectivamente, do código convolucional. O módulo da treliça  $M$  para o código consiste de  $2^m$  estados iniciais e  $2^m$  estados finais. Cada estado inicial é conectado aos estados finais por  $2^k$  ramos, e cada ramo tem  $n$  bits. A complexidade da treliça  $M$  é dada por:

$$C_M = \frac{n}{k} \cdot 2^{m+k} \quad (4.2)$$

símbolos por bit.

Seja o exemplo dado em [41] mostrado na Figura 4.6 para o código convolucional (3,2,2), onde  $n = 3$ ,  $k = 2$  e  $m = 2$ . O número total de símbolos dos ramos para cada módulo da treliça é  $3 \cdot 2^{2+2} = 48$  símbolos, ou seja  $48/2 = 24$  símbolos por bit.

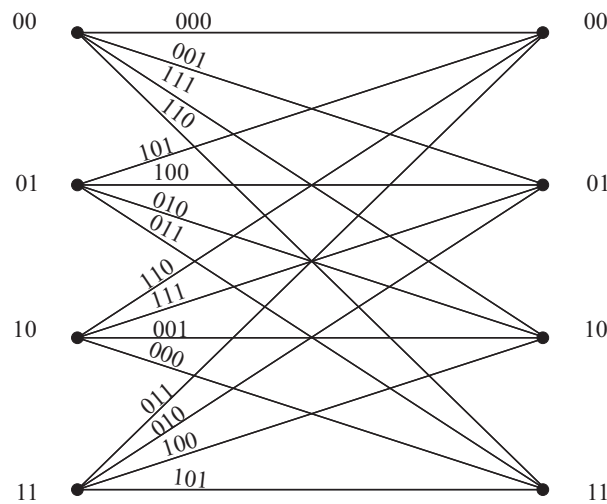


Figura 4.6: Módulo da treliça para o código convolucional (3,2,2). O número total de símbolos é 48, então a complexidade da treliça é  $48/2$  símbolos por bit

Assim, baseado nas definições dadas em [41], a complexidade de decodificação para a primeira transmissão do esquema proposto pode ser dito proporcional a:

$$C_{new}^{tx1} = \frac{n}{k} \cdot 2^{m+k}, \quad (4.3)$$

onde  $\frac{k}{n}$  é a taxa do código convolucional recursivo sistemático utilizado na primeira transmissão. Conseqüentemente, a complexidade de decodificação para a segunda, ou qualquer transmissão subsequente, é proporcional a:

$$C_{new}^{tx2} = 2 \cdot \frac{n'}{k'} \cdot i \cdot 2^{m+k}, \quad (4.4)$$

onde  $i$  é o número de iterações do decodificador turbo  $\frac{k'}{n'}$  é a taxa do código turbo depois da segunda transmissão do esquema proposto. Podemos ver da comparação de (4.3) e (4.4) que a

complexidade de decodificação aumenta consideravelmente depois da segunda transmissão. A complexidade de decodificação total para o método proposto pode ser dito proporcional a:

$$C_{new} = \left( \frac{n}{k} + N_{rtx} \cdot 2 \cdot \frac{n'}{k'} \cdot i \right) \cdot 2^{m+k}, \quad (4.5)$$

onde  $N_{rtx}$  é o número de retransmissões necessárias.

Para o esquema proposto em [34], a complexidade de decodificação é basicamente a mesma para qualquer transmissão. Usando as notações acima, podemos dizer que a complexidade de decodificação total é proporcional a:

$$C_{Nar} = N_{tx} \cdot 2 \cdot \frac{n}{k} \cdot i \cdot 2^{m+k}, \quad (4.6)$$

onde  $N_{tx}$  é o número de transmissões necessárias.

A Figura 4.7 mostra a relação  $C_{new}/C_{Nar}$  como função do número de transmissões. Os cálculos foram realizados considerando os valores  $n = 2$ ,  $k = 1$ ,  $n' = 3$ ,  $k' = 1$ , e  $m = 3$ , os quais são os valores utilizados na nas Seções 4.2 e 4.3. Pela figura podemos ver que para boas condições do canal, onde somente uma transmissão seria em geral suficiente (ou quando nenhuma retransmissão fosse necessária), o esquema proposto é consideravelmente menos complexo do que o método em [34]. Pela figura também é claro que o esquema proposto se torna mais complexo do que o método em [34] da terceira transmissão em diante.

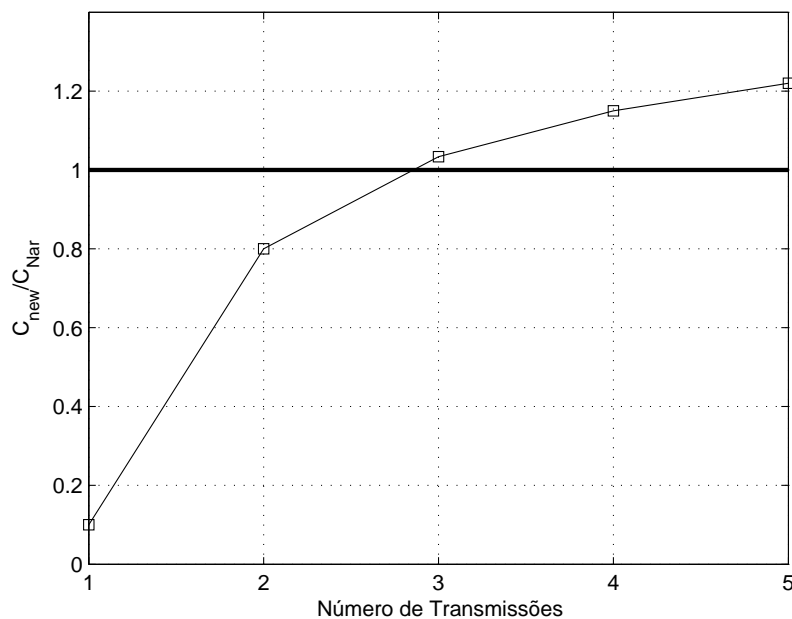


Figura 4.7: Relação de complexidade entre o esquema proposto e o método em [34],  $C_{new}/C_{Nar}$ , como função do número de transmissões, para o caso de  $n = 2$ ,  $k = 1$ ,  $n' = 3$ ,  $k' = 1$ , e  $m = 3$ .

A complexidade de decodificação total relativa à transmissão de um dado número de pacotes depende do número de retransmissões de que cada método precisa. Dado que o desempenho da  $FER$  dos dois métodos são consideravelmente diferentes, como visto na Seção 4.3, nós investigamos quantas retransmissões cada esquema precisa para a transmissão de um total de 10000 pacotes sem erro. As Figuras 4.8, 4.9, 4.10, 4.11 e 4.12 mostram, para  $E_b/N_0 = -1$  dB, 0 dB, 1 dB, 2 dB e 3 dB, quantas vezes cada esquema teve que fazer o uso da primeira, segunda, terceira, quarta e quinta transmissão. Pelas figuras podemos ver que o esquema proposto recorre à segunda transmissão um pouco mais frequentemente do que o método em [34]. Das figuras também podemos ver que o método proposto utiliza as outras transmissões subseqüentes muito menos frequentemente do que o esquema em [34].

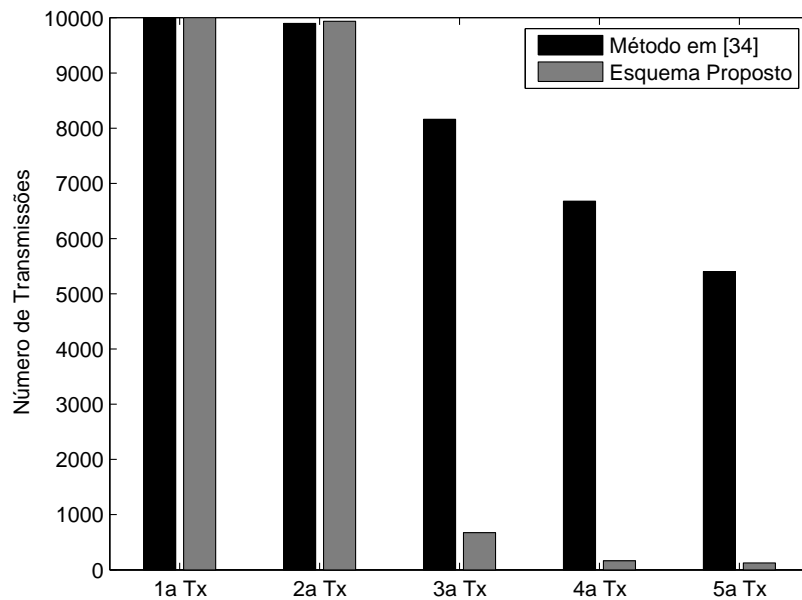


Figura 4.8: Número de vezes que o esquema proposto e o método em [34] teve que utilizar a 1<sup>a</sup>, 2<sup>a</sup>, 3<sup>a</sup>, 4<sup>a</sup> e 5<sup>a</sup> transmissão para enviar 10,000 pacotes.  $E_b/N_0 = -1$  dB.

Transmissões	Método Proposto	Método em [34]
1 <sup>a</sup> Tx	10000	10000
2 <sup>a</sup> Tx	9938	9898
3 <sup>a</sup> Tx	672	8162
4 <sup>a</sup> Tx	165	6679
5 <sup>a</sup> Tx	124	5403

Tabela 4.2: Número de vezes que o esquema proposto e o método proposto em [34] teve que utilizar a primeira, segunda, terceira, quarta e quinta transmissão, para enviar 10000 pacotes para o receptor.  $E_b/N_0 = -1$  dB.



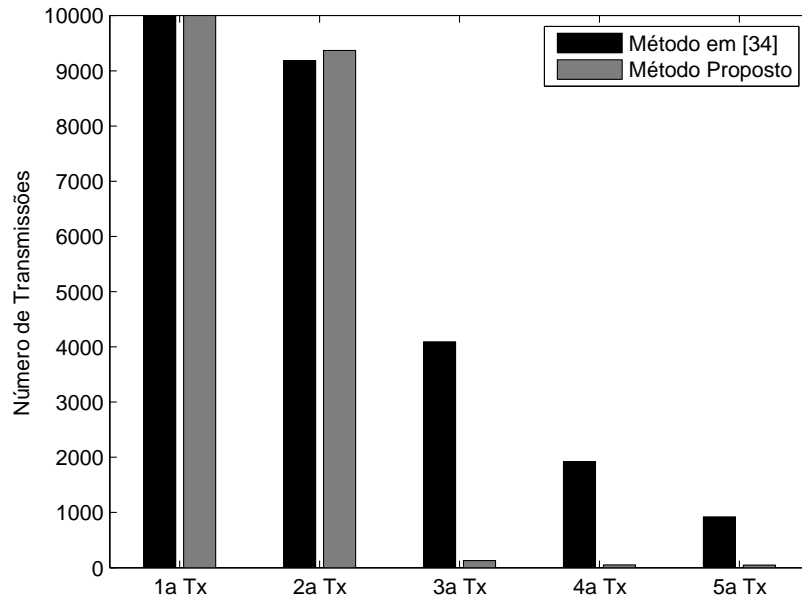


Figura 4.9: Número de vezes que o esquema proposto e o método em [34] teve que utilizar a 1<sup>a</sup>, 2<sup>a</sup>, 3<sup>a</sup>, 4<sup>a</sup> e 5<sup>a</sup> transmissão para enviar 10,000 pacotes.  $E_b/N_0 = 0$  dB.

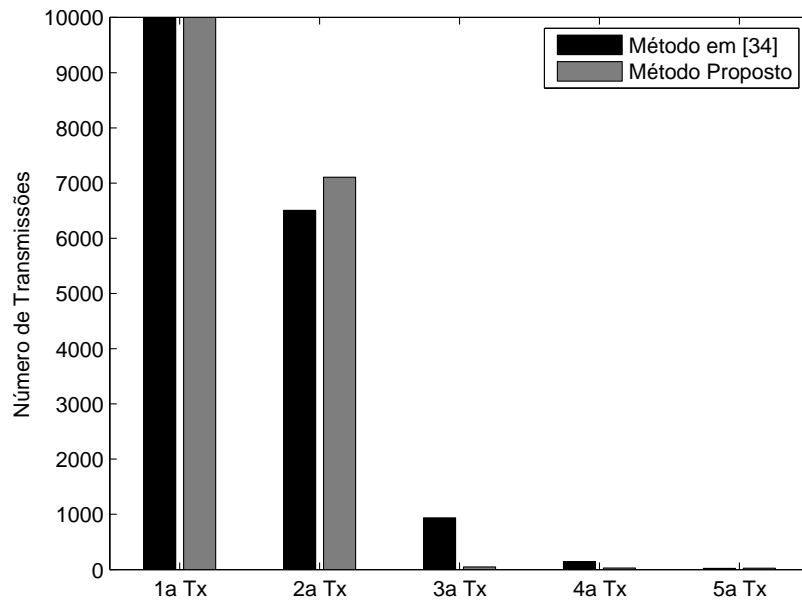


Figura 4.10: Número de vezes que o esquema proposto e o método em [34] teve que utilizar a 1<sup>a</sup>, 2<sup>a</sup>, 3<sup>a</sup>, 4<sup>a</sup> e 5<sup>a</sup> transmissão para enviar 10,000 pacotes.  $E_b/N_0 = 1$  dB.

As tabelas 4.2, 4.3, 4.4, 4.5 e 4.6 mostram quantas vezes cada esquema teve que utilizar a primeira, segunda, terceira, quarta e quinta transmissão, para os valores de  $E_b/N_0$  sob conside-

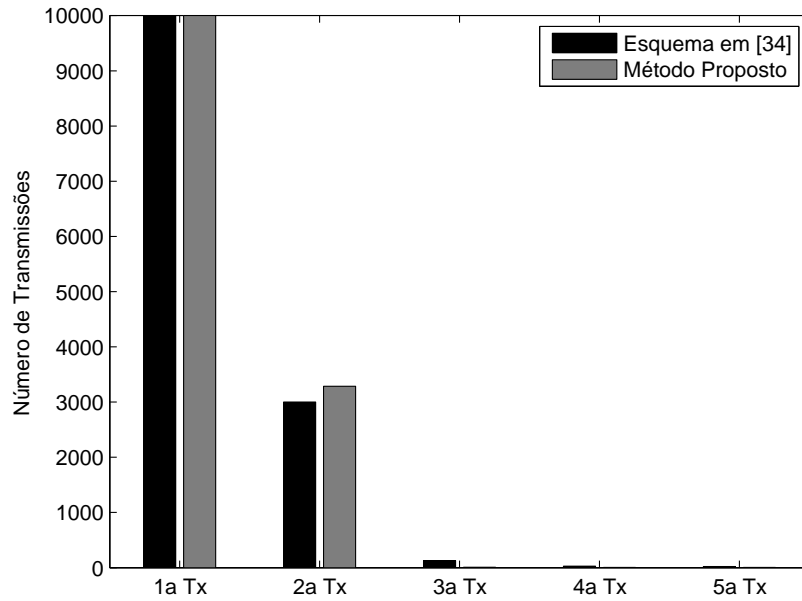


Figura 4.11: Número de vezes que o esquema proposto e o método em [34] teve que utilizar a 1<sup>a</sup>, 2<sup>a</sup>, 3<sup>a</sup>, 4<sup>a</sup> e 5<sup>a</sup> transmissão para enviar 10,000 pacotes.  $E_b/N_0 = 2$  dB.

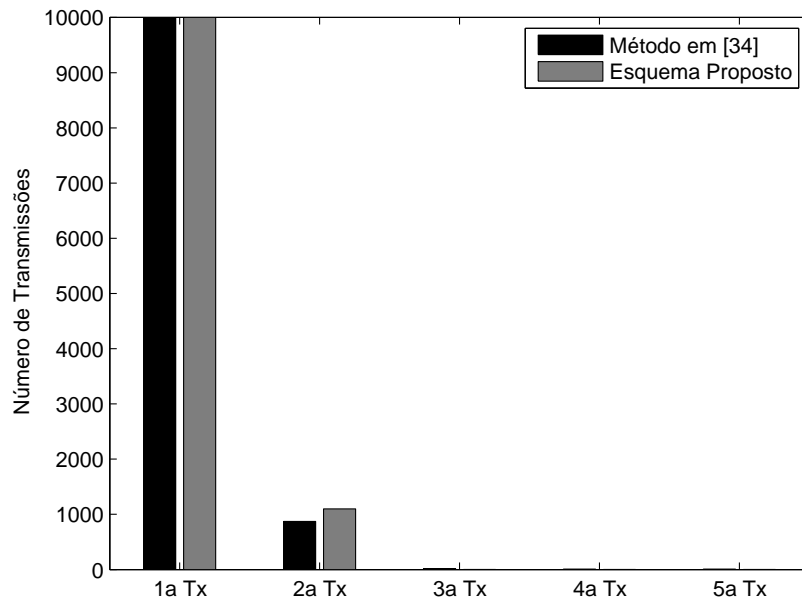


Figura 4.12: Número de vezes que o esquema proposto e o método em [34] teve que utilizar 1<sup>a</sup>, 2<sup>a</sup>, 3<sup>a</sup>, 4<sup>a</sup> e 5<sup>a</sup> transmissão para enviar 10,000 pacotes.  $E_b/N_0 = 3$  dB.

ração.

A Tabela 4.7 mostra a complexidade de decodificação total, em termos da complexidade

<b>Transmissões</b>	<b>Método Proposto</b>	<b>Método em [34]</b>
1ª Tx	10000	10000
2ª Tx	9369	9184
3ª Tx	128	4092
4ª Tx	50	1920
5ª Tx	48	921

Tabela 4.3: Número de vezes que o esquema proposto e o método proposto em [34] teve que utilizar a primeira, segunda, terceira, quarta e quinta transmissão, para enviar 10,000 pacotes para o receptor.  $E_b/N_0 = 0$  dB.

<b>Transmissões</b>	<b>Método Proposto</b>	<b>Método em [34]</b>
1ª Tx	10000	10000
2ª Tx	7107	6506
3ª Tx	47	936
4ª Tx	26	144
5ª Tx	20	21

Tabela 4.4: Número de vezes que o esquema proposto e o método proposto em [34] teve que utilizar a primeira, segunda, terceira, quarta e quinta transmissão, para enviar 10,000 pacotes para o receptor.  $E_b/N_0 = 1$  dB.

<b>Transmissões</b>	<b>Método Proposto</b>	<b>Método em [34]</b>
1ª Tx	10000	10000
2ª Tx	3287	3000
3ª Tx	9	127
4ª Tx	6	28
5ª Tx	4	20

Tabela 4.5: Número de vezes que o esquema proposto e o método proposto em [34] teve que utilizar a primeira, segunda, terceira, quarta e quinta transmissão, para enviar 10,000 pacotes para o receptor.  $E_b/N_0 = 2$  dB.

<b>Transmissões</b>	<b>Método Proposto</b>	<b>Método em [34]</b>
1ª Tx	10000	10000
2ª Tx	1011	870
3ª Tx	1	19
4ª Tx	1	8
5ª Tx	1	8

Tabela 4.6: Número de vezes que o esquema proposto e o método proposto em [34] teve que utilizar a primeira, segunda, terceira, quarta e quinta transmissão, para enviar 10,000 pacotes para o receptor.  $E_b/N_0 = 3$  dB.

da treliça, como definido em (4.5) e (4.6), e calculada baseado nos valores das tabelas 4.2, 4.3, 4.4, 4.5 e 4.6. Da tabela podemos ver que o esquema proposto requer em torno de 50% da

$E_b/N_0 = -1 \text{ dB}$		
	<b>Método Proposto</b>	<b>Método em [34]</b>
Complexidade $\infty$	5,551,520	12,845,440
$E_b/N_0 = 0 \text{ dB}$		
	<b>Método Proposto</b>	<b>Método em [34]</b>
Complexidade $\infty$	4,925,600	8,357,440
$E_b/N_0 = 1 \text{ dB}$		
	<b>Método Proposto</b>	<b>Método em [34]</b>
Complexidade $\infty$	3,776,000	5,634,240
$E_b/N_0 = 2 \text{ dB}$		
	<b>Método Proposto</b>	<b>Método em [34]</b>
Complexidade $\infty$	906,880	4,216,000
$E_b/N_0 = 3 \text{ dB}$		
	<b>Método Proposto</b>	<b>Método em [34]</b>
Complexidade $\infty$	806,204	3,489,600

Tabela 4.7: Complexidade de decodificação total, definida como em (4.5) e (4.6), para a decodificação de 10,000 pacotes, para o esquema proposto e o método em [34].

complexidade de decodificação do que o método proposto em [34] precisaria, para os casos de  $E_b/N_0 = -1 \text{ dB}$ ,  $0 \text{ dB}$  e  $1 \text{ dB}$ . De acordo com a mesma tabela, para  $E_b/N_0 = 2$  e  $E_b/N_0 = 3 \text{ dB}$  o método proposto economizaria em torno de 75% na complexidade de decodificação.

## 4.5 Comentários Finais

Neste capítulo apresentamos um novo esquema de ARQ/FEC híbrido, o qual utiliza códigos convolucionais recursivos e códigos turbo. Quando o método proposto é comparado com o popular esquema proposto por Narayanan and Stüber em [34], as seguintes conclusões podem ser feitas: i) ambos os esquemas gastam a mesma energia por uso do canal; ii) o esquema proposto alcança um desempenho melhor em termos da taxa de erro do que o esquema em [34] quando mais de uma transmissão é necessária; iii) o esquema proposto neste capítulo tem uma complexidade de decodificação menor do que aquele em [34] tanto para alta e baixa relação sinal ruído.

## 5 *Esquema de ARQ Híbrido Utilizando RSM e Códigos Turbo*

### 5.1 Introdução

Sistemas sem fios com múltiplas antenas transmissoras e/ou receptoras tem sido tópico de intensa atividade de pesquisa ultimamente. O interesse nestes sistemas com múltiplas entradas e múltiplas saídas (MIMO) foi incentivado através de dois resultados. Independentemente, Telatar [42] e Foschini e Gans [43] mostraram que pode ser obtido um significativo aumento na capacidade dos sistemas sem fio quando são empregadas múltiplas antenas no transmissor e/ou receptor. Além deste aumento da capacidade, a utilização de múltiplas antenas, pode também conduzir ao aumento da robustez em relação ao desvanecimento [20, 44], mesmo sem conhecimento do canal no transmissor. Em [14], Tarokh *et al* propuseram um esquema de transmissão conhecido como codificação espaço-temporal (STC). Nos STCs, redundância é introduzida nos fluxos de transmissão no espaço (através das antenas transmissoras) e no tempo, conduzindo ao ganho de diversidade e de codificação. STC's podem ser essencialmente divididos em *space-time trellis codes* (STTC) [14, 45] e *space-time block codes* (STBC) [13, 46].

Recentemente, uma nova estratégia de transmissão que alcança diversidade espacial baseada no mapeamento randômico do sinal (RSM) foi proposta em [17]. No RSM, o fluxo de bits é primeiramente codificado por um código corretor de erros. Então, são geradas  $N$  cópias da saída do codificador, onde  $N$  é o número de antenas transmissoras. Cada uma destas cópias passa através do mapeamento randômico do sinal, cuja saída é então transmitida através de uma antena. Além disso, o esquema com RSM tem complexidade muito menor do que os STBCs o qual é menos complexo do que os STTCs. Além do que, antenas transmissoras e/ou receptoras podem ser adicionadas no esquema do RSM sem mudanças significativas para o sistema, uma flexibilidade não encontrada nos STBCs e STTCs. Finalmente, na maioria dos STBCs, adicionar mais antenas transmissoras resulta numa redução da taxa de transmissão, redução esta que não é observada no RSM.

Alguns esquemas de ARQ híbrido usando códigos espaço-temporais podem ser encontrados na literatura [47–50], onde STTC's ou STBC's são utilizados para prover diversidade espacial. Neste capítulo nós apresentamos um novo esquema de ARQ híbrido para canais MIMO. Diversidade espacial é alcançada através do uso do RSM, evitando o aumento na complexidade com o aumento do número de antenas transmissoras típico do STC's, e a possível redução da taxa de transmissão quando mais do que duas antenas transmissoras são utilizadas, como ocorre em alguns esquemas STC's. No esquema proposto a capacidade de correção de erros é obtida através da utilização dos códigos turbo. A razão pela qual um código turbo é utilizado ao invés de um codificador mais simples (como um código convolucional) é que os códigos turbo superam o desempenho dos códigos convolucionais no canal quase-estático no caso de múltiplas antenas transmissoras [51].

Resultados de simulação investigando o desempenho do esquema proposto em termos da taxa de erro de frame (FER), *throughput* e número de retransmissões necessárias, para diferentes números de antenas transmissoras, são apresentados. Além disso, também investigamos como a melhora da camada física em um link sem fio, devido ao novo esquema de ARQ híbrido proposto, pode afetar o desempenho do protocolo da camada de transporte (TCP).

## 5.2 Modelo do Sistema

Considere a transmissão de uma sequência de bits de mensagem  $u_k$  independentes e equiprováveis, usando  $N$  antenas transmissoras. De acordo com o diagrama da Figura 5.1, os bits são primeiramente codificados pelo codificador turbo, cujo *interleaver* tem comprimento  $L$ . A saída do codificador é mapeada em  $m$ -PSK, resultando em um bloco de símbolos  $m$ -PSK  $x_k$ . Este bloco passa então através de  $N$  mapeadores aleatórios do sinal, cujas saídas são dadas por:

$$s_k^i = e^{j\phi_k^i} x_k, \quad (5.1)$$

para  $i = 1, \dots, N$ , onde  $\phi_k^i$  são  $N$  sequências independentes de fases pseudo-randômicas conhecidas no receptor. Finalmente, a sequência  $s_k^i$  é transmitida através da  $i$ -ésima antena com energia  $E_S/N$ , assim a energia total transmitida é a mesma que no caso de uma única antena.

O receptor pode empregar  $M$  antenas. O sinal recebido pela  $j$ -ésima antena receptora no instante  $k$ ,  $y^j(k)$ , é dado por:

$$y^j(k) = \sum_{i=1}^N \sqrt{E_S/N} s^i(k) h_{i,j} + \eta^j(k), \quad (5.2)$$

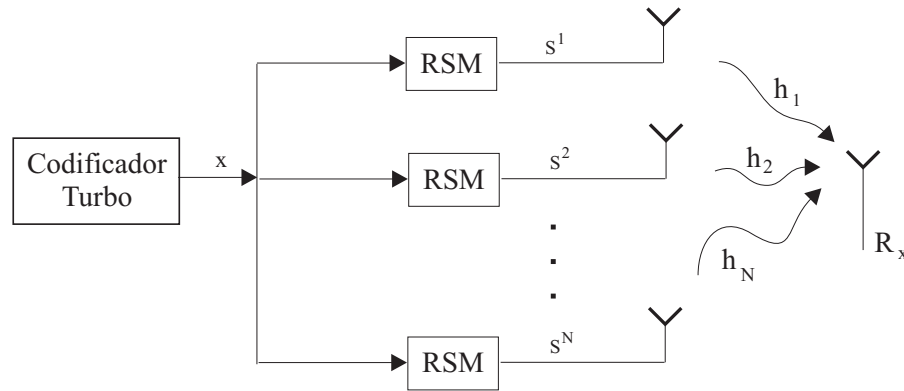


Figura 5.1: Diagrama de Blocos do sistema de transmissão com múltiplas antenas.

onde  $\eta^j(k)$  é o ruído aditivo Gaussiano com média zero e variância  $N_0/2$  por dimensão, e  $h_{i,j}$  representa o canal com desvanecimento plano entre a  $i$ -ésima antena transmissora e a  $j$ -ésima antena receptora, o qual inclui efeitos dos filtros na transmissão e recepção. Assumimos que os coeficientes  $h_{i,j}$  são variáveis aleatórias complexas Gaussianas com média zero, independentes e identialmente distribuídas, com variância unitária. Assumimos mais adiante, que os coeficientes do canal são espacialmente descorrelacionados e continuam constantes durante a transmissão de um bloco, ou uma palavra-código.

Note que usando (5.1) em (5.2) nós podemos reescrever o sinal recebido como:

$$y^j(k) = \sqrt{E_s} f_k^j x(k) + \eta^j(k), \quad (5.3)$$

onde  $f_k^j = \sqrt{1/N} \sum_{i=1}^N h_{i,j} e^{j\phi_k^i}$ . Assim, usando RSM o sinal recebido na  $j$ -ésima antena pode ser escrito como um canal com uma entrada e com uma saída (SISO) com desvanecimento  $f_k^j$  variante no tempo.

Em [17] foi mostrado que o esquema do RSM provê um ganho de diversidade que se aproxima de  $N$  com probabilidade um, em um canal com desvanecimento quase-estático, quando o comprimento efetivo  $d_v$  do código externo é muito grande ( $d_v \gg N$ ). O comprimento efetivo  $d_v$  é relacionado ao ganho de codificação do código externo. Em [17] os autores utilizaram apenas codificadores convolucionais, e mostraram que para o canal estimado o esquema proposto é mais efetivo que o esquema do Alamouti, para o caso de um codificador convolucional com alto ganho de codificação ( $v=9$ ).

Em [51] foi mostrado que, para canais com desvanecimento quase-estático, os códigos turbo não superam o desempenho dos códigos convolucionais com a mesma complexidade. Porém, em [51] também foi mostrado que para sistemas com múltiplas antenas há uma clara vantagem dos códigos turbo sobre os códigos convolucionais em relação à FER. Este resultado justifica

nossa escolha de um código turbo como o código corretor de erros do sistema de diversidade espacial mostrado na Figura 5.1.

Sem perda de generalidade, o esquema de ARQ que nós empregamos é o protocolo ARQ *stop-and-wait* [18]. Supomos que podem ser detectados erros no receptor. Na prática esta capacidade pode ser implementada através de um codificador e decodificador (CRC). Quando for detectado erro no receptor, um pedido de retransmissão é enviado ao transmissor, o qual retransmite o mesmo pacote. Quando o pacote é recebido corretamente, um sinal de reconhecimento é enviado ao transmissor, o qual transmite o próximo pacote. O canal de retorno (entre o receptor e transmissor), é assumido estar livre de ruído. Não são considerados atrasos de ida-e-volta e tempos inativos enquanto o transmissor espera pelo ACK.

### 5.3 Esquema ARQ Híbrido Usando RSM

Nesta seção investigamos o desempenho do esquema ARQ híbrido proposto. Nas simulações consideramos que os dados de entrada são codificados utilizando um codificador turbo de taxa  $1/3$  com polinômios geradores  $(21, 37)_8$  na forma octal. A modulação utilizada foi a modulação BPSK, com uma antena receptora, e *interleaver* pseudo-randômico de comprimento  $L = 97$  bits, 10 iterações no decodificador turbo, e supondo que o receptor tem conhecimento perfeito do canal.

A Figura 5.2 mostra a taxa de erro de frame (FER) em função de  $E_b/N_0$ , onde  $E_b = E_S \cdot 3$  é a energia por bit de informação, para uma, duas e quatro antenas transmissoras. Na curva também é mostrado a FER para o caso de uma antena transmissora onde o dado de entrada é codificado por um codificador convolucional de taxa  $1/3$  com polinômios geradores  $(25, 33, 37)_8$  na forma octal. Da figura podemos ver que, para o caso de uma antena transmissora, o uso do codificador turbo resulta em uma FER menor quando comparada com o codificador convolucional. Mas em contrapartida, ocorre um aumento da complexidade. Porém, ganhos maiores podem ser obtidos quando mais antenas transmissoras são empregadas. Para uma FER de  $10^{-2}$  a  $E_b/N_0$  necessária é de 23, 20, 13 e 9 dB para o caso do código convolucional com uma antena transmissora, o codificador turbo com uma antena transmissora, o codificador turbo com duas antenas transmissoras e codificador turbo com quatro antenas transmissoras, respectivamente. A maior parte desta melhora é devido à adição da segunda antena transmissora. Aumentando ainda mais o número de antenas transmissoras resulta em um ganho muito pequeno, em relação ao caso de quatro antenas transmissoras.

A Figura 5.3 mostra a FER em função de  $E_b/N_0$ , para uma e duas antenas transmissoras



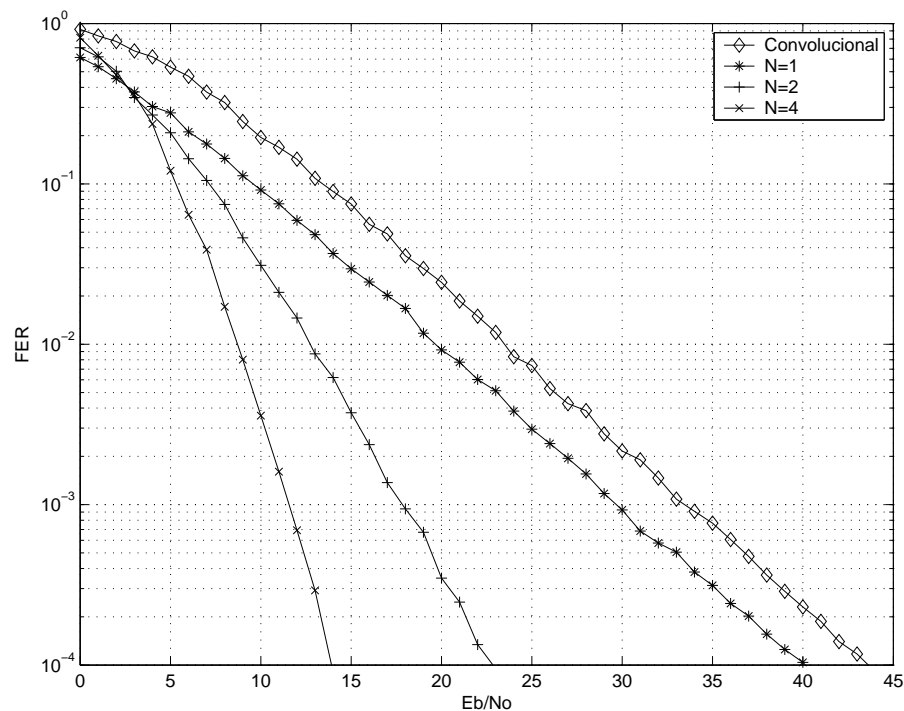


Figura 5.2: Taxa de erro de frame (FER) por  $E_b/N_0$  para um código turbo de taxa 1/3 com uma, duas e quatro antenas transmissoras usando mapeamento randômico do sinal. Na curva também é mostrado a FER para o caso de uma antena transmissora e um código convolucional de taxa 1/3.

usando mapeamento randômico do sinal e o código turbo de taxa 1/3, depois da primeira e segunda transmissões. Pela figura podemos notar a grande melhora na FER devido à introdução do protocolo ARQ. Considerando que a realização de canal é descorrelacionada de uma transmissão para outra, a probabilidade de um desvanecimento profundo ocorrer duas vezes para o mesmo pacote de dados é muito baixa. Para uma FER de  $10^{-2}$  são necessários  $E_b/N_0$  de 7 e 5 dB para os casos de uma e duas antenas transmissoras, respectivamente. Ocorreu uma melhora de 4 e 2 dB, respectivamente, quando comparado com o caso de apenas uma transmissão.

A Figura 5.4 mostra o efeito do esquema proposto no número de retransmissões necessárias para transmissão de 100,000 pacotes com erro zero, considerando  $E_b/N_0 = 15$  dB. Da figura podemos ver que o número de transmissões necessárias é altamente reduzido pelo uso de múltiplas antenas transmissoras. Especificamente, para o caso do codificador convolucional e uma antena transmissora são necessárias 107169 transmissões (7169 retransmissões, ou 7.169 % a mais), enquanto que para o caso do codificador turbo e quatro antenas transmissoras são necessárias só 100072 transmissões (apenas 72 retransmissões, ou só 0.072 % a mais). Esta redução tem um impacto considerável no tempo de vida útil de uma bateria de dispositivos com potência limitada e no *throughput* dos protocolos das camadas superiores, como nós discutimos na próxima seção.

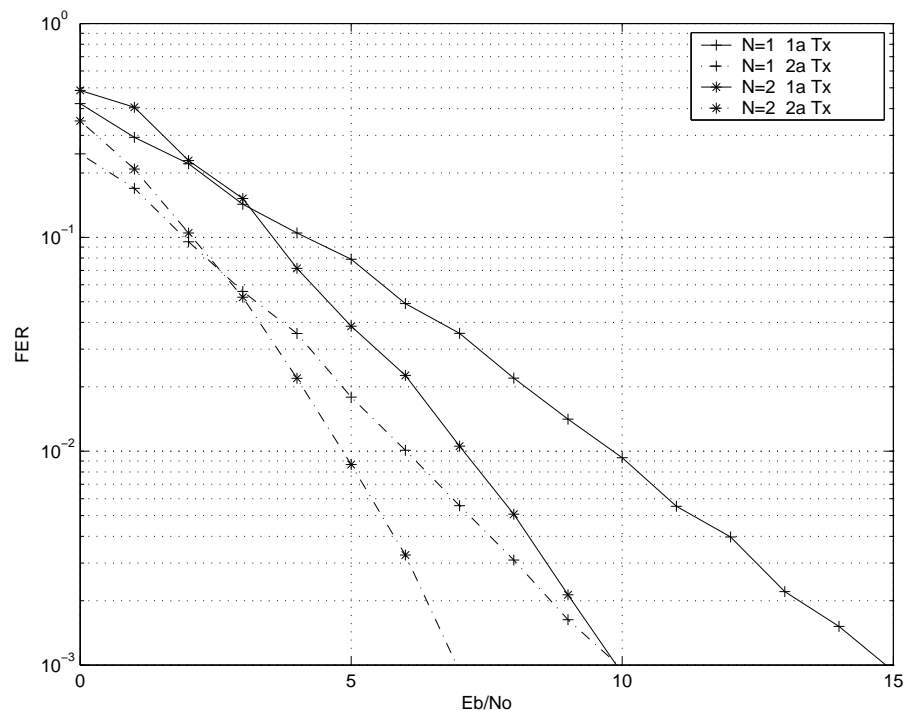


Figura 5.3: Taxa de erro de frame (FER) depois da primeira e da segunda transmissões por  $E_b/N_0$  para um código turbo de taxa 1/3 com uma e duas antenas transmissoras usando mapeamento randômico do sinal.

## 5.4 Throughput do TCP sobre Links Sem fio Usando RSM e Códigos Turbo

O protocolo TCP (Transport Control Protocol) é um protocolo de transporte fim-a-fim responsável pela transmissão de cerca de 90% do tráfego da internet [52]. Ele apresenta um bom desempenho nos canais com fio onde a taxa de erros é baixa e a perda de pacotes é devido ao congestionamento na rede. Neste caso, quando a informação chegar ao destino incorreta ou se ela não chegar no destino depois do *time out*, o TCP envia um ACK negativo para o transmissor e inicia um mecanismo para o controle do congestionamento.

Jacobson propôs em [53] o uso da janela de congestionamento dinâmica onde seu tamanho pode variar de acordo o estado da rede. Este mecanismo para controle do congestionamento consiste em duas fases, a primeira chamada de *slow start* e a segunda *congestion avoidance*. Na primeira fase a janela do TCP cresce exponencialmente e na segunda fase a janela do TCP cresce linearmente [54]. Quando for detectado o congestionamento a janela do TCP se reduz drasticamente. Este mecanismo dinâmico permite resolver o congestionamento rapidamente e ainda usar eficientemente a largura de banda disponível na rede.

Existem diferentes versões do TCP, como Vegas, Reno e New Reno. No primeiro sempre

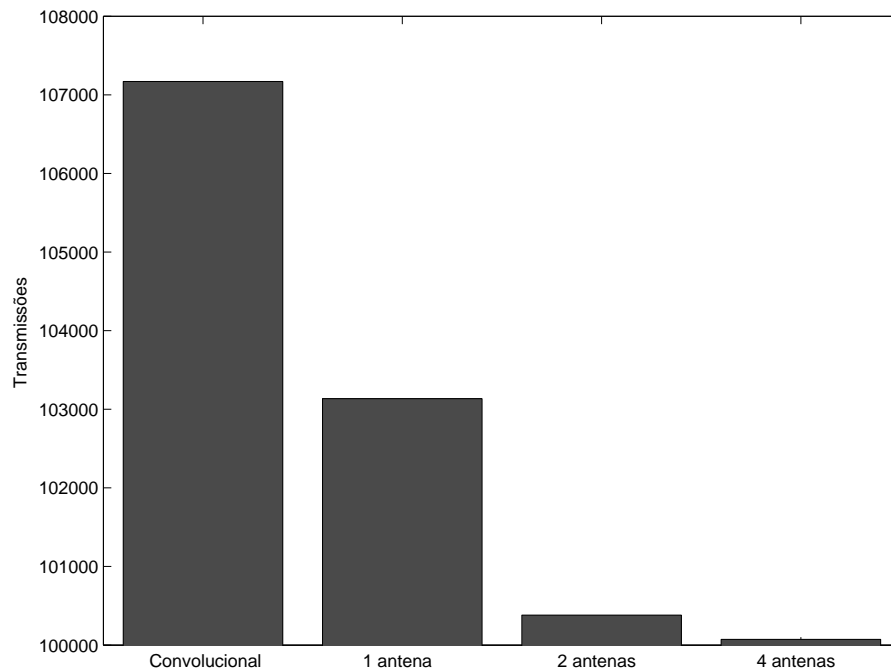


Figura 5.4: Número total transmissões para comunicação com erro zero de 100,000 pacotes, para o caso de um código turbo de taxa  $1/3$  com uma, duas e quatro antenas transmissoras usando mapeamento randômico do sinal. Na figura também é mostrado o número total de transmissões para o caso de uma antena transmissora e de um código convolutacional de taxa  $1/3$ . Com o valor de  $E_b/N_0 = 15$  dB para os quatro casos mostrados na figura.

que uma perda é detectada a janela reduz seu tamanho ao valor um e a fase *slow start* inicia. Neste caso há uma drástica diminuição no tamanho da janela e conseqüentemente na taxa de transmissão. No TCP tipo Reno e New Reno a janela se reduz somente a um se a perda é detectada depois do *time-out*. Quando a perda é detectada durante os ACKs a janela de congestionamento se reduz à metade. A fase *slow start* não é inicializada e o protocolo continua na fase de controle de congestionamento [52].

Nos canais sem fio, onde a ocorrência de erros é muito mais freqüente, a perda de pacotes devido às altas taxas de erros são mais significativas do que a perda de pacotes devido ao congestionamento [55]. Neste caso, o TCP aciona constantemente o mecanismo para o controle de congestionamento diminuindo a largura de banda e degradando o *throughput*. Desta forma, é necessário modificar o protocolo para levar em consideração as características dos links sem fio, acionando o mecanismo de controle de congestionamento somente quando necessário [54].

Recentemente, vários esquemas foram propostos para combater os efeitos da degradação no desempenho do TCP sobre os links sem fio. Em [56] Balakrishnan *et al* compararam alguns esquemas projetados para melhorar o desempenho do TCP em tais redes. Eles classificaram estes esquemas em três grandes categorias: protocolos *end-to-end*, onde a recuperação da perda

é executada pelo remetente; protocolos *link-layer*, os quais fornecem confiabilidade local; e protocolos *split-connection* os quais rompem a conexão fim-a-fim em duas partes na estação base.

Nesta seção nós investigamos como o desempenho do *throughput* do TCP na camada de transporte pode ser consideravelmente melhorado aumentando apenas a confiabilidade do link sem fio. O aumento da confiabilidade pode ser obtido pelo uso do esquema FEC/ARQ híbrido proposto.

### 5.4.1 Avaliação do Throughput do TCP Usando o Esquema ARQ Híbrido

O desempenho do *throughput* do TCP usando RSM foi simulado com o simulador de redes NS2. Consideramos a implementação das três versões do TCP citadas acima, TCP Reno, New Reno e Vegas. A taxa de transmissão no canal foi especificada em 1Mbps e o tamanho dos quadros da camada física foi especificado como sendo 256 Bytes. A aplicação utilizada sobre o TCP foi o serviço de transferência de arquivos (FTP). Consideramos quatro esquemas para a camada física, chamados de CS1, CS2, CS3 e CS4. CS1 utiliza o código convolucional e uma antena transmissora. CS2 utiliza o código turbo e uma antena transmissora. CS3 utiliza o código turbo, duas antenas transmissoras e RSM. CS4 utiliza o código turbo, quatro antenas transmissoras e RSM. O código turbo e o código convolucional utilizados foram os definidos na Seção 5.3.

A topologia da simulação está mostrada na Figura 5.5, onde o nó N1 representa o esquema com o codificador convolucional e uma antena transmissora (CS1). Os nós N2, N3 e N4 correspondem ao esquema com o codificador turbo com uma, duas e quatro antenas transmissoras, respectivamente CS2, CS3 e CS4. O nó N5 representa o roteador e N6 o nó de destino. Os nós N1, N2, N3 e N4 estão conectados com o nó N5 usando um link bidirecional com 10ms de atraso de propagação e capacidade de 1 Mbps. O nós N5 e N6 também estão conectados com um link bidirecional com 10ms de atraso de propagação e capacidade de 10Mbps.

A camada física é modelada usando os resultados de FER obtidas na simulação mostrada na Figura 5.4, onde consideramos o ponto de operação como sendo  $E_b/N_0 = 15dB$ . O desempenho do *throughput* na camada de transporte é mostrado na Figura 5.6, considerando TCP New Reno, onde as variações nas curvas são devido à perda de pacotes.

Pela figura podemos ver que o esquema que utiliza o código convolucional, CS1, apresenta o pior desempenho. O desempenho do sistema melhora quando o codificador turbo é utilizado ao invés do codificador convolucional, como pode ser visto na curva CS2.

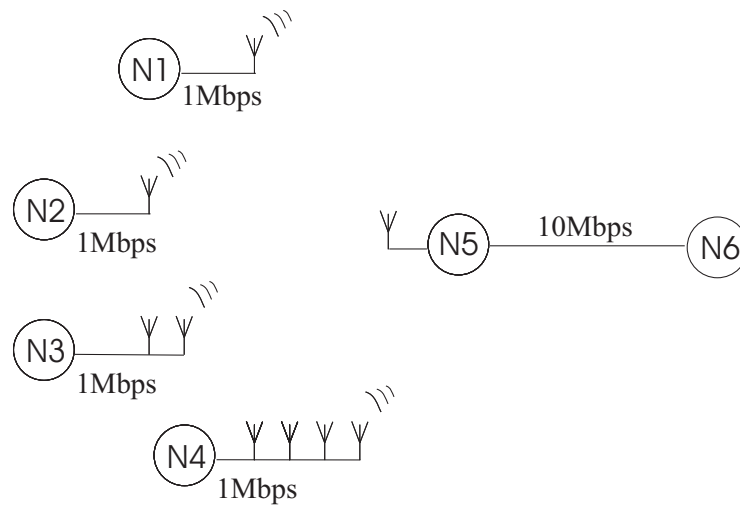


Figura 5.5: Topologia da simulação.

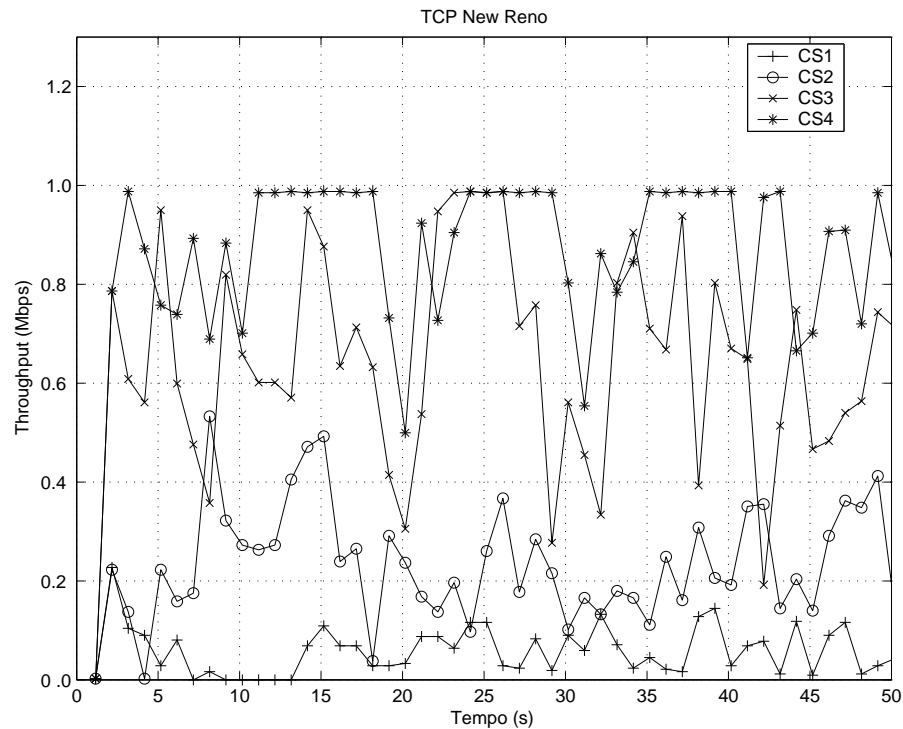


Figura 5.6: Throughput do TCP sobre quatro diferentes configurações de links sem fio, considerando TCP New Reno.

Uma grande melhora no *throughput* é obtida quando diversidade espacial é utilizada usando RSM com duas antenas transmissoras (curva CS3). Com quatro antenas o desempenho do TCP é ainda melhor, como mostra a curva CS4. Calculando o valor médio do *throughput* com 50s de simulação para as quatro configurações obteve-se o seguinte resultado: 52.25 kbps para uma antena transmissora e o código convolucional (CS1); 228.63 kbps para uma antena transmissora e o código turbo (CS2); 643.1 kbps para duas antenas transmissoras usando RSM e o código

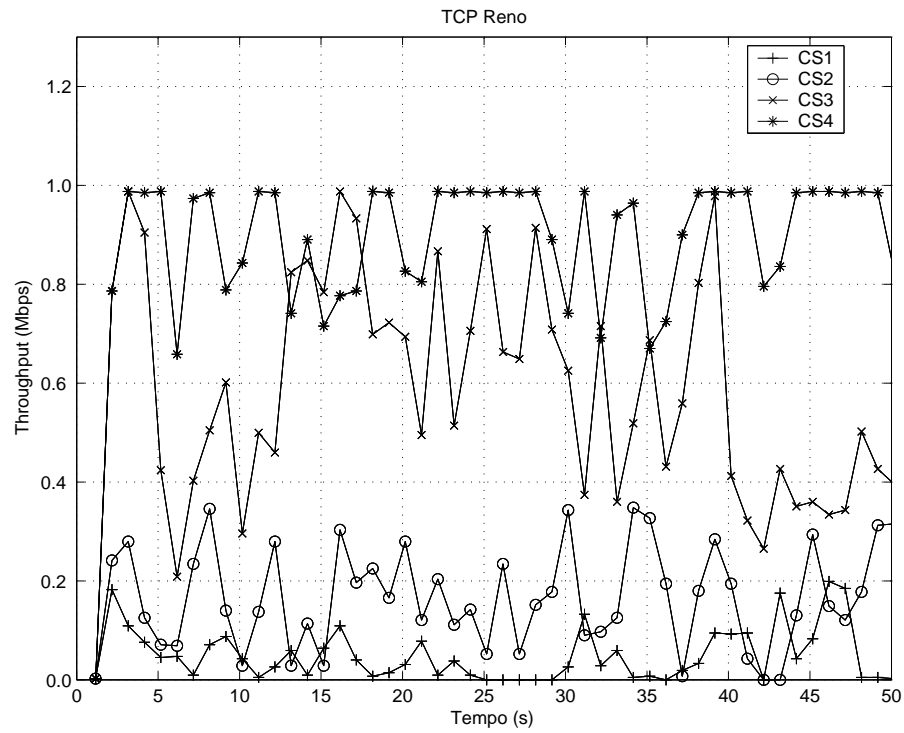


Figura 5.7: Throughput do TCP sobre quatro diferentes configurações de links sem fio, considerando TCP Reno.

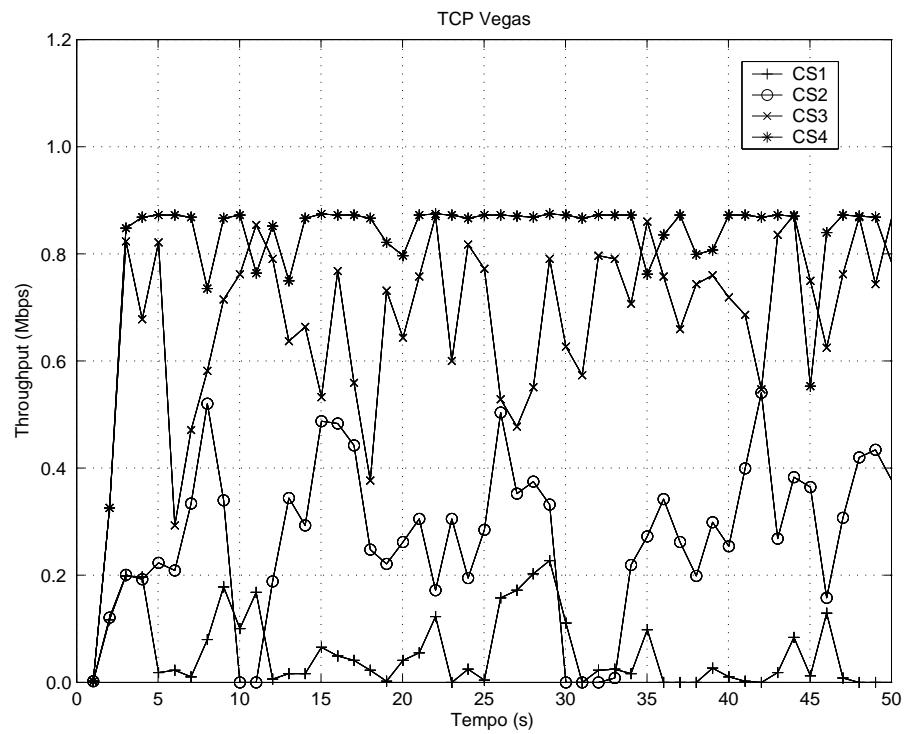


Figura 5.8: Throughput do TCP sobre quatro diferentes configurações de links sem fio, considerando TCP Vegas.

turbo (CS3); e 860.4 kbps para quatro antenas transmissoras usando RSM e o código turbo (CS4). O ganho no desempenho do TCP para o caso de CS4 sobre CS2 é mais do que 250%.

Resultados semelhantes foram obtidos para o TCP tipo Reno, mostrado na Figura 5.7 e para o TCP Vegas mostrado na Figura 5.8. O *throughput* médio com os esquemas CS1, CS2, CS3 e CS4 para o TCP tipo Reno e Vegas está mostrado na Tabela 5.1. Para o TCP tipo Reno *throughput* médio foi de: 49.20 kbps para o esquema (CS1); 195.42 kbps para o esquema (CS2); 583.71 kbps para o esquema (CS3); e 884.78.4 kbps para o esquema (CS4). O ganho no desempenho do TCP para o caso de CS4 sobre CS2 é em torno de 352%. Para o TCP tipo Vegas, obteve-se os valores de: 56.62 kbps para o esquema (CS1); 268.86 kbps para o esquema (CS2); 661.66 kbps para o esquema com (CS3); e 807.68 kbps para o esquema com (CS4). O ganho no desempenho para o caso de CS4 sobre CS2 é em torno de 200%.

	<b>Throughput médio (kbps)</b>			
<b>TCP</b>	CS1	CS2	CS3	CS4
Reno	49.20	195.42	583.71	884.78
Vegas	56.62	268.86	661.66	807.68

Tabela 5.1: Valor médio do Throughput das quatro configurações para o TCP Reno e Vegas.

As Tabelas 5.2, 5.3 e 5.4 mostram o número de pacotes enviados, recebidos e perdidos para cada nó da rede, considerando as três implementações de TCP (Reno, New Reno e Vegas). Podemos ver uma considerável diferença na taxa de perda quando comparamos os nós N4 e N3 (os quais exploram diversidade espacial) sobre os nós N2 e N1 (os quais não exploram diversidade espacial). Das tabelas torna-se claro que o esquema proposto pode melhorar consideravelmente a o desempenho de um link sem fio.

	<b>Reno</b>			
<b>Número de pacotes</b>	Nó N1	Nó N2	Nó N3	Nó N4
Enviados	1120	3568	12371	18693
Recebidos	1039	3493	12325	1868
Perdidos	81	75	46	11
Taxa de perda	7.23%	2.10%	0.37%	0.058%

Tabela 5.2: Taxa de perda para os nós N1, N2, N3 e N4, considerando a implementação do TCP tipo Reno e diferentes configurações na camada física.

	<b>New Reno</b>			
<b>Número de pacotes</b>	Nó N1	Nó N2	Nó N3	Nó N4
Enviados	1334	5024	13617	18182
Recebidos	1230	4924	13578	18168
Perdidos	104	100	39	14
Taxa de perda	7.79%	1.99%	0.28%	0.077%

Tabela 5.3: Taxa de perda para os nós N1, N2, N3 e N4, considerando a implementação do TCP New Reno e diferentes configurações na camada física.

	<b>Vegas</b>			
<b>Número de pacotes</b>	Nó N1	Nó N2	Nó N3	Nó N4
Enviados	1510	6704	16523	19971
Recebidos	1410	6564	16477	19957
Perdidos	100	140	46	14
Taxa de perda	6.66%	2.09%	0.28%	0.070%

Tabela 5.4: Taxa de perda para os nós N1, N2, N3 e N4, considerando diferentes a implementação do TCP Vegas e diferentes configurações na camada física.

## 5.5 Comentários Finais

Neste capítulo nós apresentamos um esquema FEC/ARQ híbrido, o qual alcança ganho de codificação e de diversidade. Diversidade espacial é obtida através do mapeamento randômico do sinal em cada antena, enquanto que ganho de codificação é alcançado através da codificação turbo. Resultados de simulação mostraram melhoras significativas em relação a FER e *throughput* que podem ser obtidas com o aumento do número de antenas transmissoras. No esquema proposto não ocorre o aumento da complexidade ou redução da taxa de transmissão típico dos outros sistemas ARQ espaço-temporais. Mostramos também que a utilização da codificação e diversidade através da codificação turbo e do RSM, podem consideravelmente melhorar o desempenho do TCP sobre os links sem fio.



## 6 *Conclusão e Trabalhos Futuros*

Nesta dissertação nós propomos dois esquemas de ARQ híbrido utilizando o princípio da codificação turbo. No Capítulo 4 foi apresentado um esquema de ARQ híbrido para o canal SISO e no Capítulo 5 foi apresentado um esquema de ARQ híbrido para canal MISO.

O primeiro esquema proposto utiliza a técnica de combinação de pacotes. Na primeira transmissão os bits são codificados por um codificador convolucional recursivo sistemático de taxa  $1/2$ , na segunda transmissão os bits são codificados por um codificador convolucional recursivo de taxa 1. Depois da segunda transmissão os bits da primeira transmissão são utilizados em conjunto com os bits da segunda transmissão (técnica de combinação de pacotes), operando assim como um código turbo de taxa  $1/3$  no receptor. Este esquema foi comparado com o esquema proposto por Narayanan e Stüber [34], em relação à taxa de erro de frame (FER) e *throughput*. Nossos resultados mostraram que o esquema proposto alcança uma melhor taxa de erro de frame e *throughput*. Além disso, ambos os métodos gastam a mesma energia para transmitir um determinado número de bits de informação. A complexidade do dois esquemas foi comparada considerando o número de retransmissões que cada esquema precisaria para a transmissão de um total de 10,000 pacotes. Consideramos neste caso ao todo cinco transmissões, os resultados mostraram que o método proposto tem uma complexidade de decodificação menor do que o outro método avaliado.

O segundo esquema proposto utiliza a codificação turbo e diversidade espacial. Resultados de simulação mostraram que significativas melhoras da taxa de erro de frame e *throughput* podem ser obtidas com o aumento do número de antenas transmissoras. Comparado com outros esquemas de diversidade espacial, o RSM é mais simples e mais flexível ao aumento de antenas transmissoras e/ou receptoras, e principalmente, não reduz a taxa de transmissão do sinal ao adicionar mais antenas, como ocorrem nos outros esquemas. Mostramos também através de simulações, que o esquema proposto pode consideravelmente melhorar o desempenho do TCP sobre os links sem fio em uma camada superior.

Como trabalhos futuros, o esquema proposto no Capítulo 4 para o canal AWGN SISO pode

ser feito para o Canal Rayleigh com desvanecimento rápido e também para o Canal Rayleigh com desvanecimento lento, neste último caso a técnica da diversidade espacial pode ser aplicada para melhorar ainda mais a qualidade do sinal recebido. Outros métodos já feitos para estes dois tipos de canais podem ser analisados e implementados, para que os novos esquemas sugeridos possam desta forma ser avaliados.

## *Referências Bibliográficas*

- [1] W. Pawel and S. Michal, "The First Steps in Telecommunications". Disponível em: [www2.hs-esslingen.de/telehistory/1ststeps.htm](http://www2.hs-esslingen.de/telehistory/1ststeps.htm). Acessado em 29/11/2006.
- [2] Corazza, G.C. Marconi's history. Proceedings of the IEEE, vol. 86, n. 7, pp 1307-1317, July, 1998.
- [3] C.E. Shannon, A Mathematical Theory of Communication, Bell System Tech. J. 27 (1948), 379–423, 623–656. MR MR0026286 (10,133e)
- [4] C.E. Shannon, Communication in the presence of noise, Proceedings of the IRE, Vol. 37, 1949, 10-21.
- [5] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding Turbo Codes- Proc. Int. Conf. Communications, Maio 1993pp, 1064-1070.
- [6] Ioannis A. Chatzigeorgiou, Miguel R.D. Rodrigues, Ian J. Wassell and Rolando Carrasco, "A Comparison of Convolutional and Turbo Coding Schemes for Broadband FWA Systems", 12th International Conference on Telecommunications, Cape Town, South Africa, May 2005.
- [7] Bahl, L. R., Cocke, J., Jelinek, F., and Raviv, J., "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," in IEEE Trans. Information Theory, March 1974, pp. 284-287.
- [8] J. G. Proakis, Digital Communications, Mc Graw Hill, New York, 4th edition, 2000.
- [9] Guimarães, Dayani Adionel. Introdução às Comunicações Móveis. Disponível em : [www.inatel.br/revista/volume-01/pg1.htm](http://www.inatel.br/revista/volume-01/pg1.htm). Acessado em: 20/10/2006.
- [10] S. William: Wireless Communications and Networks, Prentice Hall, ISBN 0-13-040864-6, 2002.
- [11] A. Paulraj, R. Nabar, D. Gore: Introduction to Space-Time Wireless Communications, Published May 2003, ISBN:0521826152
- [12] H. Bolcskei and A. J. Paulraj, "Multiple-input multiple-output (MIMO) wireless systems", chapter in "The Communications Handbook", 2nd edition, J. Gibson, ed., CRC Press, pp. 90.1 - 90.14, 2002.
- [13] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1451–1458, Oct. 1998.
- [14] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communications: performance criterion and code construction," *IEEE Trans. Inform. Theory*, vol. 44, no. 2, pp. 744–765, Mar. 1998.

- [15] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Transaction on Information Theory*, vol. 45, pp.1456-1467, July 1999.
- [16] W. Su and X.-G. Xia, "Two generalized complex orthogonal space-time block codes of rates 7/11 and 3/5 for 5 and 6 transmit antennas," presented as recent results in the International Symposium on Information Theory, Washington DC, June 2001.
- [17] Y. Li, C. N. Georghiades, and Garng Huang, "Transmit diversity over quasi-static fading channels using multiple antennas and random signal mapping," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1918–1926, Nov. 2003.
- [18] S. Lin, D. J. Costello Jr., and M. J. Miller, "Automatic-repeat-request error-control schemes," *IEEE Commun. Mag.*, vol. 22, no. 12, pp. 5-17, Dec. 1984.
- [19] H. Liu, H. Ma, M. El Zarki, and S. Gupta, "Error control schemes for networks: An overview," *Mobile Networks Applicat.*, vol. 2, no. 2, pp. 167–182, 1997.
- [20] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice-Hall, 1996.
- [21] D. Divsalar and F. Pollara, "Multiple turbo codes for deep-space communications," JPL TDA Progress Report, pp. 66-77, May 1995.
- [22] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, Englewood Cliffs: Prentice Hall, 1995
- [23] P. Sindhu, "Retransmission error control with memory," *IEEE Trans. Commun.*, vol. 25, no. 5, pp. 473-479, May 1977.
- [24] D. Chase, "Code-combining - a maximum likelihood decoding approach for combining an arbitrary number of noisy packets," *IEEE Trans. on Commun.*, vol. 33, May 1985.
- [25] Harvey, B. A. and Wicker, S. B., "Packet Combining Systems Based on the Viterbi Decoder," *Proceedings of the 1992 IEEE Military Communications Conference*, pp. 757-761, San Diego, California, October 11-14, 1992.
- [26] J. Hagenauer. Rate-compatible punctured convolutional codes (RCPC codes) and their applications. *IEEE Transactions On Communications*, 36(4):389–400, April 1988.
- [27] A. Burr, "Turbo-codes: the ultimate error control codes?" *Electronics and Communication Engineering Journal*, August, 2001, pp. 155-165.
- [28] D. J. Costello, Jr., H. A. Cabral, and O. Y. Takeshita, "Some thoughts on the equivalence of systematic and nonsystematic convolutional encoders," *Forney Fest Program of Events*, pp. 1-2, American Academy of Arts and Sciences, Cambridge, MA, March 2000.
- [29] B. Vucetic and J. Yuan, *Turbo Codes: Principles and Applications* New York: Kluwer Academic, 2000, pp. 211-213.
- [30] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm "in *IEEE Trans. Information Theory*, vol. IT-13, April 1967 , pp. 284-287.
- [31] William E. Ryan, "A Turbo Code Tutorial". Disponível em: <http://www.ece.arizona.edu/ryan/>

- [32] I. Land and P. A. Hoeher, "Using the mean reliability as a design and stopping criterion for turbo codes," in Proc. IEEE Information Theory Workshop, Sept. 2001, pp. 27-29.
- [33] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261-1271, Oct. 1996.
- [34] K. R. Narayanan and G. L. Stüber, "A novel ARQ technique using the turbo coding principle," *IEEE Commun. Letters*, vol. 1, no. 2, pp. 49-51, Mar. 1997.
- [35] Morelos-Zaragoza, Robert H., "The Art of ECC". Disponível em: <http://the-art-of-ecc.com>.
- [36] J. Chakareski and P. Chou, "Application layer error-correction coding for rate-distortion optimized streaming to wireless clients," *IEEE Trans. Commun.*, vol. 52, no. 10, pp. 1675-1687, Oct. 2004.
- [37] R. Liu, P. Spasojevic, and E. Soljanin, "A throughput analysis of incremental redundancy hybrid ARQ schemes with turbo codes," *Proc. Conference on Information Sciences and Systems (CISS)*, Mar. 2004.
- [38] V. Tripathi, E. Visotsky, R. Peterson, and M. Honig, "Reliability-based type II hybrid ARQ schemes," *Proc. IEEE ICC*, May 2003.
- [39] M. M. Butt, J. Ch. Fricke, and P. A. Hoeher, "Reliability-based packet combining with application to interleave-division multiple access," *Proc. Int. Symp. on Turbo Codes*, Apr. 2006.
- [40] Y. Wu and M. C. Valenti, "An ARQ technique using related parallel and serial concatenated convolutional codes," *Proc. IEEE ICC*, June 2000.
- [41] R. J. McEliece and W. Lin, "The trellis complexity of convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 1855-1864, Nov. 1996.
- [42] E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Transactions on Telecommunications*, pp. 585-595, Nov. 1999.
- [43] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Pers. Commun.*, vol. 6, pp. 311-335, Mar. 1998.
- [44] E. Biglieri, J. Proakis, and S. Shamai, "Fading channels: information-theoretic and communications aspects," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2619-2692, Oct. 1998.
- [45] A. R. Hammons Jr. and H. El Gamal, "On the theory of space-time codes for PSK modulation," *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 524-542, Mar. 2000.
- [46] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1456-1467, July 1999.
- [47] W. T. Kim, S. J. Bae, J. G. Kim, and E. K. Joo, "Performance of STBC with turbo code in HARQ scheme for mobile communication systems," *ICT 2003*, vol. 1, pp. 85-89, Mar. 2003.

- [48] M.-K. Oh, Y.-H. Kwon, and D.-J. Park, "Efficient hybrid ARQ with space-time coding and low-complexity decoding," *IEEE ICASSP 2004*, vol. 4, pp. iv-589-iv-592, May 2004.
- [49] Y. S. Jung and J. H. Lee, "Hybrid-ARQ scheme employing different space-time trellis codes in slow fading channels," *IEEE VTC 2002-Fall*, vol. 1, pp. 247-251, Sep. 2002.
- [50] A. V. Nguyen and M. A. Ingram, "Hybrid ARQ protocols using space-time codes," *IEEE VTC 2001-Fall*, vol. 4, pp. 2364-2368, Oct. 2001.
- [51] M. R. D. Rodrigues, I. Chatzgeorgiou, I. J. Wassel, and R. Carrasco, "On the performance of turbo codes in quasi-static fading channels," *IEEE ISIT 2005*, pp. 622-626, Sep. 2005.
- [52] E. Altman, T. Jimenez, "NS simulator for beginners," Lecture notes, autumn 2002, Univ. de. Los Andes, Merida, Venezuela, July 2003.
- [53] V. Jacobson and K. J. Michael, "Congestion avoidance and control", *Proc. ACM SigComm*, pp. 314-329, 1988.
- [54] A. C. F. Chan, D. H. K. Tsang and S. Gupta, "TCP (Transmission Control Protocol) over wireless links," *Proc. VTC'97*, pp. 2326-2330, Phoenix, U.S.A., May 1997.
- [55] H. Bai, D. Lija, M. Atiquzzaman, "Improving TCP throughput over lossy links using protocol level speculations," *Proc. IEEE Wirelesscom*, Maui, HI, June 2005.
- [56] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Trans. Net.*, vol. 5, no. 6, Dec. 1997.

# Resumo

Este trabalho apresenta dois esquemas de ARQ híbrido utilizando o princípio da codificação turbo. O primeiro esquema, foi proposto para o canal SISO, e o segundo esquema para o Canal MIMO. O primeiro esquema de ARQ híbrido utiliza códigos convolucionais recursivos e o princípio da decodificação turbo. Na primeira transmissão os dados são codificados por um codificador convolucional sistemático recursivo. Na segunda transmissão (primeira retransmissão) os dados são entrelaçados ou embaralhados por um *interleaver* e codificados por um codificador convolucional recursivo. Depois da segunda transmissão o esquema proposto opera como um decodificador turbo. O método proposto é comparado com um esquema conhecido na literatura, ambos em termos do desempenho da taxa de erro, *throughput* e complexidade de decodificação. Foi considerado que ambos os métodos gastam a mesma energia para transmitir um determinado número de bits de informação. O segundo esquema de ARQ híbrido para canal MIMO, utiliza a codificação turbo tradicional e uma técnica simples de diversidade de transmissão, o mapeamento randômico do sinal (RSM). O esquema foi avaliado com diferentes números de antenas transmissoras em relação à taxa de erro, número de retransmissões necessárias e por último, avaliamos como o esquema proposto na camada física pode afetar o *throughput* do Protocolo da Camada de Transporte (TCP).

## PALAVRAS-CHAVE

Códigos Turbo, FEC, ARQ, HARQ, RSM.

## ÁREA / SUB-ÁREA DE CONHECIMENTO

3.00.00.00-9 – Engenharia

3.04.06.005 – Telecomunicações

2006

Nº: 433

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)



[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)