



TRABALHO DE GRADUAÇÃO

Simulação dos algoritmos
de Rate Matching e HARQ
no enlace direto do LTE

Lucas Sousa e Silva

Brasília, dezembro de 2014

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

**Simulação dos algoritmos
de Rate Matching e HARQ
no enlace direto do LTE**

Lucas Sousa e Silva

*Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Engenheiro Eletricista*

Banca Examinadora

Prof. André Noll Barreto, ENE/UnB

Orientador

Prof. Marcelo Menezes de Carvalho, ENE/UnB

Examinador interno

Prof. João Paulo Leite, ENE/UnB

Examinador interno

Agradecimentos

Agradeço à todos que direta ou indiretamente me ajudaram a realizar este trabalho.

Lucas Sousa e Silva

RESUMO

O padrão LTE e sua versão melhorada LTE Advanced estão estabelecendo-se cada vez mais como o sistema dominante para as redes de telecomunicações móveis de quarta geração. O objetivo deste trabalho é estudar o padrão LTE, especificamente o processo de *rate matching* e os protocolos HARQ feitos no *Downlink Shared Channel* (DL-SCH).

Para alcançar os objetivos traçados utilizou-se o simulador de enlace direto das camadas física e de enlace WiSiL, que foi desenvolvido pela Universidade de Brasília.

Ao longo do texto são expostas de forma sucinta as características técnicas do padrão LTE para melhor compreender os resultados da simulação.

Os resultados da simulação de *rate matching* mostraram que a variação da taxa de codificação permite que sejam utilizadas técnicas de modulação e codificação adaptativas que conferem maior eficiência espectral ao sistema.

A simulação de HARQ mostrou que os protocolos de HARQ são capazes de reduzir consideravelmente a taxa de bits errados em cada retransmissão e são, por conseguinte, capazes de fornecer um serviço eficiente de retransmissão de pacotes ao sistema.

ABSTRACT

LTE and its enhanced version LTE Advanced are establishing themselves as the increasingly dominant standard for the fourth generation mobile telecommunication networks. The objective of this work is to study the LTE standard, specifically the Downlink Shared Channel (DL-SCH) rate matching process and HARQ protocols.

To achieve the stated objectives, the link-level downlink simulator WiSiL was used, which was developed by the University of Brasilia.

Throughout the text some specifications of the LTE standard are briefly exposed in order to better comprehend the simulation results.

The rate matching simulation results showed that flexibility in coding rate allows the implementation of adaptive modulation and coding techniques that provide greater spectral efficiency.

The results of HARQ simulations showed that the HARQ protocols are capable of substantially reducing the bit error rate in each retransmission and are therefore able to provide an efficient packet retransmission service to the system.

SUMÁRIO

1	Introdução	1
1.1	HISTÓRICO DOS SISTEMAS DE COMUNICAÇÃO CELULARES	1
1.1.1	ALÉM DO 3G	2
1.1.2	EVOLUÇÃO AO 4G	3
1.2	OBJETIVOS	3
1.3	ESTRUTURA DO TRABALHO	5
2	O padrão LTE	6
2.1	INTRODUÇÃO	6
2.2	CARACTERÍSTICAS TÉCNICAS FUNDAMENTAIS DO LTE	6
2.2.1	OFDM	6
2.2.2	MIMO	8
2.3	CANAIS DO LTE	8
2.4	ESTRUTURA DO FRAME LTE E GRADE DE RECURSOS	11
3	CRC e Códigos Turbo	14
3.1	CRC	14
3.2	CÓDIGOS TURBO	15
3.2.1	INTRODUÇÃO	15
3.2.2	CÓDIGOS CONVOLUCIONAIS	15
3.2.3	CÓDIGOS CONCATENADOS E CODIFICADOR TURBO	21
3.2.4	DECISÃO RÍGIDA E DECISÃO SUAVE (<i>hard-decision</i> E <i>soft-decision</i>)	23
3.2.5	LLR	25
3.2.6	DECODIFICAÇÃO ITERATIVA	26
4	O bloco de codificação de canal e HARQ	28
4.1	O BLOCO DE CODIFICAÇÃO DE CANAL DO LTE	28
4.1.1	CRC	28
4.1.2	SEGMENTAÇÃO EM BLOCOS	28
4.1.3	CODIFICAÇÃO TURBO	29
4.1.4	RATE MATCHING	30
4.2	HARQ	32
4.2.1	STOP-AND-WAIT	32

4.2.2	GO-BACK-N	33
4.2.3	SELECTIVE REPEAT	34
4.2.4	N-CHANNEL STOP-AND-WAIT	36
4.2.5	TIPOS DE HARQ	37
5	Simulação.....	40
5.1	O SIMULADOR WiSiL.....	40
5.2	RESULTADOS	42
5.2.1	VARIAÇÃO DA TAXA DE CODIFICAÇÃO	42
5.2.2	HARQ.....	44
6	Conclusões.....	50
6.1	CONSIDERAÇÕES FINAIS.....	50
6.2	TRABALHOS FUTUROS.....	50
	REFERÊNCIAS BIBLIOGRÁFICAS	52

LISTA DE FIGURAS

1.1	Evolução dos sistemas de comunicação celulares [3].	4
2.1	As camadas do sistema LTE [2].	9
2.2	Mapeamento entre diferentes tipos de canais [2].	11
2.3	<i>Radio frame</i> do LTE [2].	11
2.4	Grade de recurso [2].	13
3.1	Codificador convolucional [6].	16
3.2	Codificador convolucional $k = 1$, $K = 3$ e $n = 3$ [6].	17
3.3	Codificadores Equivalentes [6].	19
3.4	Diagrama de Estados [6].	20
3.5	Diagrama de Treliça [6].	21
3.6	Codificador Sistemático Recursivo [6].	22
3.7	Codificador Turbo LTE [5].	23
3.8	Sistema de comunicação [9].	23
3.9	Funções de verossimilhança [9].	25
3.10	Decodificação Iterativa [9].	27
4.1	Bloco de codificação do LTE [3].	29
4.2	As 3 etapas de rate matching [3].	31
4.3	ARQ <i>Stop-and-wait</i> [3].	33
4.4	ARQ <i>Go-back-N</i> [3].	34
4.5	ARQ <i>Selective Repeat</i> [3].	35
4.6	Janelas de recepção e transmissão do ARQ <i>Selective Repeat</i> [3].	36
4.7	ARQ <i>N-channel stop-and-wait</i> [3].	37
5.1	Diagrama de Blocos do simulador WiSiL [13].	40
5.2	Ber em função da SNR por <i>bit</i> para várias taxas de codificação - canal AWGN.	44
5.3	Ber em função da SNR por <i>bit</i> para várias taxas de codificação - canal 3GPP SCM.	44
5.4	BER em função da SNR por <i>bit</i> para 1 retransmissão (ou seja sem HARQ) e 2 retransmissões HARQ tipo 1.	46
5.5	HARQ tipo 1 - BER em função da SNR por <i>bit</i> para sucessivas retransmissões.	46
5.6	Ber em função da SNR por <i>bit</i> para 1 retransmissão (ou seja, sem HARQ) e 2 retransmissões HARQ tipo 3.	47
5.7	HARQ tipo 3 - BER em função da SNR por <i>bit</i> para sucessivas retransmissões.	48

5.8	Comparação entre HARQ tipo 1 com <i>chase combining</i> e HARQ tipo 3.....	48
-----	--	----

LISTA DE TABELAS

2.1	Resumo das principais características técnicas do padrão LTE.	7
5.1	Parâmetros mantidos constantes durante a simulação de <i>rate matching</i>	43
5.2	Parâmetros mantidos constantes durante a simulação HARQ.	45

LISTA DE SÍMBOLOS

Símbolos

K	Comprimento de bits limitantes
R	Taxa de codificação
s_r	Símbolo recebido corrompido por ruído
s_t	Símbolo transmitido
\hat{s}_t	Estimação do símbolo transmitido
$L(s_t)$	LLR <i>a priori</i> do símbolo s_t
$L_e(\hat{s}_t)$	LLR extrínseca
$L'(\hat{s}_t)$	LLR após demodulação
$L(\hat{s}_t)$	LLR após decodificação

Lista de Siglas

1G	Primeira geração
2G	Segunda geração
3G	Terceira geração
4G	Quarta geração
5G	Quinta geração
3GPP	<i>3rd Generation Partnership Project</i>
3GPP2	<i>3rd Generation Partnership Project 2</i>
ACK	<i>Acknowledgment</i>
APP	<i>A Posteriori Probability</i>
ARQ	<i>Automatic Repeat Request</i>
AWGN	<i>Additive White Gaussian Noise</i>
BCH	<i>Broadcast Channel</i>
BCCH	<i>Broadcast Control Channel</i>
BER	<i>Bit Error Rate</i>
BPSK	<i>Binary phase-shift keying</i>
CCCH	<i>Common Control Channel</i>
CDMA	<i>Code Division Multiple Access</i>
CoMP	<i>Coordinated Multi-Point</i>
CRC	<i>Cyclic Redundancy Check</i>
DCCH	<i>Dedicated Control Channel</i>
DL-SCH	<i>Downlink Shared Channel</i>
DTCH	<i>Dedicated Traffic Channel</i>
EUTRA	<i>Evolved Universal Terrestrial Radio Access</i>
FDD	<i>Frequency Division Duplexing</i>
FFT	<i>Fast Fourier Transform</i>
GSM	<i>Global System for Mobile communications</i>
HARQ	<i>Hybrid Automatic Repeat reQuest</i>
HRPD	<i>High Rate Packet Data</i>
HSPA	<i>High Speed Packet Access</i>
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IFFT	<i>Inverse Fast Fourier Transform</i>
IS-95	<i>Interim Standard 95</i>
ISI	<i>Intersymbol Interference</i>
IUT	<i>International Telecommunication Union</i>

LAN	<i>Local Area Network</i>
LDPC	<i>Low Density Parity Codes</i>
LLR	<i>Log Likelihood Ratio</i>
LMSC	<i>LAN/MAN Standard Comitee</i>
LTE	<i>Long Term Evolution</i>
LRT	<i>Likelihood Ratio Test</i>
MAC	<i>Medium Access Control</i>
MAN	<i>Metropolitan Area Network</i>
MIMO	<i>Multiple Input Multiple Output</i>
NACK	<i>Negative Acknowledgment</i>
OFDM	<i>Orthogonal Frequency Division Multiplexing</i>
OFDMA	<i>Orthogonal Frequency Division Multiple Access</i>
OSI	<i>Open Systems Interconnection</i>
PDCP	<i>Packet Data Convergence Protocol</i>
PDSCH	<i>Physical Downlink Shared Channel</i>
QAM	<i>Quadrature Amplitude Modulation</i>
QoS	<i>Quality of Service</i>
QPSK	<i>Quadrature Phase-shift Keying</i>
RLC	<i>Radio Link Control</i>
RRC	<i>Radio Resource Control</i>
RSC	<i>Recursive Systematic Convolutional</i>
SAW	<i>Stop-and-Wait</i>
SFBC	<i>Space-frequency block coding</i>
SNR	<i>Signal-to-Noise Ratio</i>
SCFDMA	<i>Single Carrier Frequency Division Multiplexing</i>
SIHO	<i>Soft Input/Hard Output</i>
SISO	<i>Soft Input/Soft Output</i>
TDD	<i>Time Division Duplexing</i>
TTI	<i>Transmission Time Interval</i>
UMB	<i>Universal Mobile Broadband</i>
WCDMA	<i>Wideband Code Division Multiple Access</i>
WiMAX	<i>Worldwide Interoperability for microwave access</i>
WiSiL	<i>Wireless Simulation Laboratory</i>
WLAN	<i>Wide Area Networks</i>

Capítulo 1

Introdução

Ao longo das últimas décadas, a indústria de comunicações celulares sem fio viveu crescimento estrondoso. Segundo estimativas da ITU (*International Telecommunication Union*), ao fim de 2014 o número de assinantes de serviços de telecomunicações móveis celulares vai quase atingir 7 bilhões, o que corresponde a uma taxa de penetração de 96% [1].

Para entender as causas da ubiquidade dos sistemas de telecomunicação celulares é preciso analisar seu histórico. Os primeiros sistemas de telecomunicações celulares surgiram no final da década de 1970 e no início da década de 1980 nos Estados Unidos, Japão e em parte da Europa [2]. O serviço era apenas de voz e limitado, mas desde o seu surgimento, a taxa de crescimento de usuários de sistemas de telecomunicações celulares sempre foi elevada. Em menos de 20 anos, o número de assinantes cresceu de zero a mais de um bilhão [2]. O crescimento elevado foi movido pelo desejo das pessoas de se conectarem e compartilharem informação em trânsito, e o desenvolvimento desta tecnologia só foi possível graças aos avanços nas áreas de processamento de sinais e microeletrônica [2].

Apesar de a telefonia móvel ter sido o propulsor inicial dos sistemas de telecomunicações sem fio (e ainda é a aplicação primária), desde a última década o serviço de transmissão de dados tem um papel cada vez mais importante no setor. Nos últimos vinte anos, a Internet evoluiu muito, e se tornou uma rede de informação indispensável, provedora de uma vasta gama de serviços.

Neste contexto de demanda cada vez maior por mais altas taxas de transmissão de dados é que se insere o padrão LTE, cuja origem é exposta de forma sucinta neste capítulo [2].

1.1 Histórico dos sistemas de comunicação celulares

A primeira geração (1G) de sistemas de telecomunicações celulares tem como característica marcante a utilização de modulação analógica: é a única geração em que técnicas de modulação analógicas são utilizadas. Os sistemas de primeira geração foram projetados para prestar serviço de telefonia móvel com itinerância restrita [3]. O serviço ficava muitas vezes restrito a um país ou a uma região devido à presença de muitos padrões, o que gerava incompatibilidades entre sistemas

diferentes [2].

A segunda geração (2G) de sistemas de telecomunicações celulares passou a empregar esquemas de modulação digital e prometia maior capacidade (mais usuários) e melhor qualidade de voz que o sistema analógico de primeira geração. Além disso, a itinerância passou a ser mais simples, principalmente na Europa, devido à adoção de um único padrão e à alocação de faixas de espectro comuns, gerando um cenário mais homogêneo que nos sistemas 1G. Os dois sistemas dominantes na segunda geração eram o GSM (*global system for mobile communications*) e o IS-95 (*interim standard 95*) [3]. Assim como na primeira geração, os sistemas 2G tinham uma arquitetura de rede baseada na comutação de circuitos e seu foco era oferecer serviço de comunicação de voz. Por isso, o suporte a dados era muito limitado, possuindo taxas de transmissão máximas de apenas algumas centenas de quilobits por segundo [2].

Com a maturação dos sistemas 2G, a ITU estabeleceu metas para alavancar a evolução para a terceira geração (3G) de sistemas de comunicação celulares. Entre os requisitos propostos pela ITU estão uma taxa de pico de transmissão de dados de 2Mbit/s e suporte para mobilidade veicular. Surgiu o 3GPP, (*3rd generation partnership project*) formado por grupos de telecomunicações interessados em desenvolver um padrão de terceira geração baseado na tecnologia GSM que fosse compatível com os requisitos da ITU. Surgiu também o 3GPP2 (*3rd generation partnership project 2*), que se propunha a desenvolver um padrão concorrente ao herdeiro do GSM baseado na tecnologia IS-95 utilizada na segunda geração. O 3GPP produziu o padrão WCDMA (*wideband code division multiple Access*) e o 3GPP2, o padrão cdma2000 [3].

Graças aos esforços dos dois 3GPPs, tanto o WCDMA, como o cdma2000 cumpriram a promessa de oferecer as altas taxas de transmissão de dados exigidas pela ITU para os sistemas de terceira geração. Visando melhorar seus sistemas de terceira geração e atender à crescente demanda por taxas de transmissão mais altas, o 3GPP e o 3GPP2 desenvolveram, respectivamente, as famílias de protocolos HSPA (*high speed packet access*) e o HRPD (*high rate packet data*). O HSPA e o HRPD complementavam o WCDMA e o cdma2000, respectivamente, e adicionavam melhorias ao tráfego de dados como adaptação rápida de enlace e HARQ (*hybrid automatic repeat request*) [3].

1.1.1 Além do 3G

Enquanto o HSPA e o HRPD eram desenvolvidos e implementados, o IEEE 802 LMSC (*LAN/MAN standard comitee*) desenvolveu o padrão IEEE 802.16e, que descrevia um sistema capaz de prover acesso banda larga móvel com taxas de transmissão e eficiência espectral superiores aos sistemas de terceira geração em atuação [3]. O IEEE 802.16e era um melhoramento e uma extensão do padrão IEEE 802.16 que foi concebido para acesso banda larga fixo sem fio. Ele se destacava dos padrões de terceira geração por empregar OFDMA (*orthogonal frequency division multiple access*) em vez do CDMA utilizado nos sistemas 3G. Apesar de o nome oficial da família de normas e especificações IEEE 802.16 ser WirelessMAN, a tecnologia ficou conhecida pelo nome de WiMAX (*worldwide interoperability for microwave access*) devido ao WiMAX Forum, grupo formado por empresas atuantes na indústria de telecomunicações que tinha por objetivo promover

o uso do WiMAX e garantir a compatibilidade e interoperabilidade de produtos baseados nesta tecnologia [3].

O surgimento do WiMAX levou o 3GPP e o 3GPP2 a desenvolverem seus próprios padrões para além da terceira geração. O 3GPP2 desenvolveu o *universal mobile broadband* (UMB) e o 3GPP desenvolveu o *evolved universal terrestrial radio access* (EUTRA), também chamado de LTE (*long term evolution*). Os 3 sistemas (WiMAX, UMB e LTE) utilizam OFDMA no enlace direto e um modelo de arquitetura similar, baseado em uma rede de comutação de pacotes. Além disso, os três padrões permitem o uso de bandas flexíveis, esquemas MIMO (*multiple input multiple output*) e FDD (*frequency division duplexing*) ou TDD (*time division duplexing*) [3].

Em 2008, a Qualcomm, principal desenvolvedor do UMB, anunciou que abandonaria o projeto devido à pouca adesão ao padrão pelas operadoras de telecomunicações móveis [4]. Dessa maneira, apenas o WiMAX e o LTE são atualmente utilizados como sistemas pré-quarta geração baseados em OFDM, com o LTE dominando o mercado.

O LTE e WiMAX são comercializados como sistemas de quarta geração, mas na realidade eles não o são, pois não cumprem os requisitos de um sistema 4G determinados pela ITU em sua especificação IMT-A (*international mobile telecommunications advanced*). Apesar disso, tanto LTE como WiMAX apresentam drásticas mudanças em relação às tecnologias empregadas na terceira geração, justificando serem classificados separadamente [2].

1.1.2 Evolução ao 4G

Em 2011, o IEEE 802 LMSC (*LAN/MAN standard comitee*) e o 3GPP lançaram, respectivamente, o IEEE 802.16.m e o *LTE Advanced* - versões melhoradas de seus padrões, que atingem os requisitos de um sistema 4G determinados pela ITU. Entre esses requisitos estão uma taxa de transmissão média no enlace direto de 100 Mbit/s em redes de longa distância (*wide area networks*) e até 1 Gbit/s para redes de acesso local e em cenários de baixa mobilidade [3]. O objetivo desses novos padrões é melhorar ainda mais a eficiência espectral e aumentar as taxas de transmissão, mantendo retrocompatibilidade com os sistemas anteriores. A Figura 1.1 mostra um diagrama da evolução dos sistemas de comunicação celulares ao longo das gerações.

1.2 Objetivos

O foco deste trabalho está no estudo do padrão LTE, com ênfase nos algoritmos de *rate matching* e HARQ.

O *rate matching* é o processo de adequação da taxa de codificação de canal. É feito dentro do contexto de modulação e codificação adaptativas. A adaptação e codificação adaptativas é o nome dado a variação automática do esquema de modulação e da taxa de codificação de canal de acordo com a qualidade do canal. Se o canal está com boa qualidade, usa-se uma taxa de codificação alta e um esquema de modulação de ordem elevada. Se o canal está com má qualidade, usa-se uma taxa de codificação baixa e um esquema de modulação de ordem pequena. Graças a modulação e

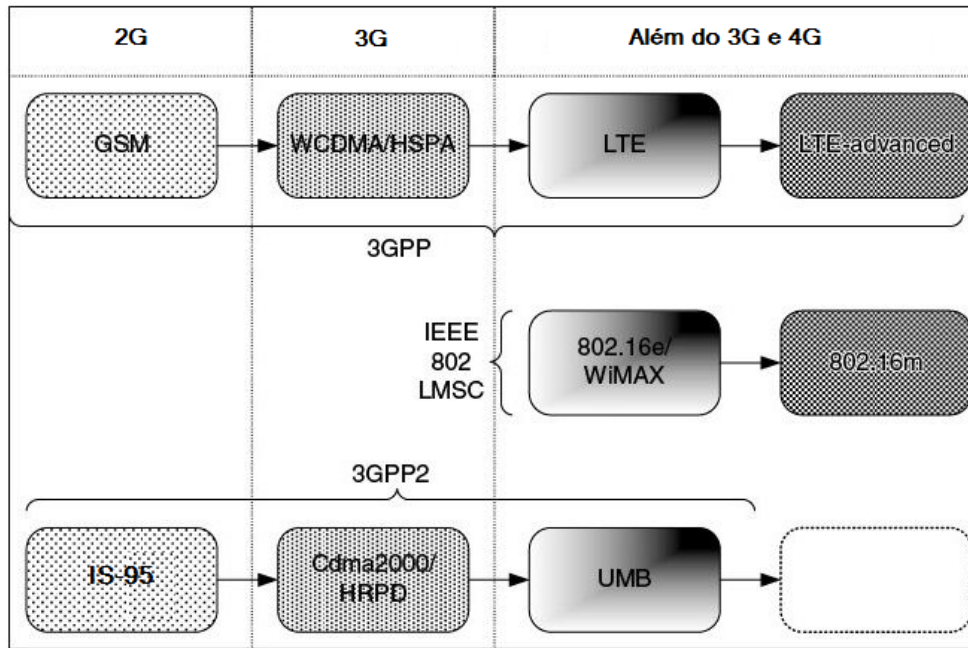


Figura 1.1: Evolução dos sistemas de comunicação celulares [3].

codificação adaptativas, é possível maximizar as taxas de transmissão de dados.

O HARQ é um protocolo de retransmissão de pacotes usado na camada MAC do padrão LTE para reduzir a taxa de erros de pacotes. Seu uso permite reaproveitar pacotes corrompidos para aumentar a chance de decodificação correta do pacote retransmitido.

Neste trabalho serão abordados em detalhe o funcionamento tanto do processo de *rate matching*, como do HARQ dentro do contexto do padrão LTE.

O simulador WiSiL (*wireless simulation laboratory*) é um simulador de enlace direto da camada física e MAC do LTE, desenvolvido pela Universidade de Brasília. Seu desenvolvimento é feito principalmente por alunos em trabalhos de conclusão de curso. Pouco a pouco, com a contribuição de muitas pessoas, o simulador torna-se cada vez mais completo. No início deste trabalho, o WiSiL já contava com diversas funcionalidades, e as etapas de codificação/decodificação, modulação, estimação de canal, montagem do quadro LTE e outras já estavam completas ou quase completas.

Na etapa de codificação de canal, faltava implementar o *rate matching*. Sem o *rate matching* não é possível variar a taxa de codificação, de modo que faltava esta funcionalidade ao simulador.

Além do *rate matching*, o HARQ, intimamente ligado ao *rate matching*, não funcionava corretamente. O HARQ era outra importante função que faltava ao simulador WiSiL.

O objetivo deste trabalho é desenvolver essas duas funcionalidades e, usando o simulador WiSiL, realizar uma simulação dos algoritmos de *rate matching* e HARQ no enlace direto do padrão LTE.

1.3 Estrutura do Trabalho

No segundo capítulo, O Padrão LTE, são expostas as características do sistema LTE, com foco para os aspectos mais relevantes para o entendimento da simulação.

No terceiro capítulo, CRC e códigos turbo, é abordado o funcionamento dos códigos detectores e corretores de erro utilizados no LTE.

No quarto capítulo, o bloco de codificação de canal e HARQ, é descrito com detalhes o bloco de codificação de canal do LTE, onde ocorre o *rate matching*. A seguir é explicado sucintamente o funcionamento do HARQ no LTE.

No quinto capítulo, Simulação, é explicado o funcionamento do simulador WiSiL e o trabalho que foi desenvolvido. São analisados gráficos relativos à simulação de *rate matching* e HARQ.

No último capítulo, Conclusão, os resultados são lembrados e as conclusões mais pertinentes são apresentadas.

Capítulo 2

O padrão LTE

Neste capítulo são expostas, em linhas gerais, algumas características técnicas do padrão LTE e são detalhados os pontos mais importantes para a compreensão dos capítulos subsequentes.

2.1 Introdução

A proliferação dos *smartphones* e a disponibilidade de serviços de acesso à banda larga móvel, fruto dos avanços conquistados na terceira geração, possibilitaram aos usuários desfrutar de toda a riqueza multi-mídia que a Internet pode oferecer. O padrão LTE surge para suprir a crescente demanda por tráfego de dados em sistemas de telecomunicação sem fio móveis. Os objetivos do LTE são: oferecer um desempenho comparável ao acesso à banda larga cabeada a usuários de redes celulares móveis, coexistir e interoperar com os diferentes padrões de terceira geração já existentes e permitir uso flexível de espectro para facilitar a implementação do sistema pelas operadoras de redes móveis. Para atingir esses objetivos, foi necessário fazer grandes mudanças em relação à arquitetura da rede e às técnicas de rádio enlace usadas em gerações passadas.

Graças à utilização de OFDMA no enlace direto, e SC-FDMA (*single carrier frequency division multiplexing*) no enlace de subida, o sistema LTE permite o uso de bandas flexíveis. O emprego de técnicas de MIMO ou antenas inteligentes permite aumentar consideravelmente as taxas de transmissão, melhorando a eficiência espectral do sistema. A Tabela 2.1 resume as principais características técnicas do padrão LTE.

O foco do trabalho está nas etapas de codificação e decodificação de canal, onde são feitos os procedimentos de *rate matching* e HARQ.

2.2 Características técnicas fundamentais do LTE

2.2.1 OFDM

Um dos grandes desafios para transmissões de alta taxa de *bits* em um canal sem fio é a interferência entre símbolos causada pelo multipercuro. Ao se propagar, as ondas de rádio estão sujeitas

Largura de Banda	1,25-20 MHz
Duplexing	FDD, TDD, <i>half duplex</i> FDD
Mobilidade	350 km/h
Múltiplo Acesso (Enlace direto)	OFDMA
Múltiplo Acesso (Enlace de subida)	SC-FDMA
MIMO (Enlace direto)	2 x 2, 4 x 2, 4 x 4
MIMO (Enlace de subida)	1 x 2, 1 x 4
Taxa máxima de transmissão de dados (Enlace direto)	173 e 326 Mbit/s para MIMO 2 x 2 e 4 x 4
Taxa máxima de transmissão de dados (Enlace de subida)	86 Mbit/s com MIMO 1 x 2
Modulação	QPSK, 16 QAM, 64QAM
Codificação de canal	Turbo
Outras técnicas	Adaptação de enlace, controle de potência Alocação de recursos sensível ao canal e HARQ

Tabela 2.1: Resumo das principais características técnicas do padrão LTE.

a reflexões no solo e em obstáculos. O sinal chega então ao receptor por diversos caminhos, e no receptor tem-se uma superposição do sinal transmitido. Devido às ondas percorrerem diferentes distâncias, o sinal transmitido e seus 'ecos' estão defasados entre si.

Quando o atraso entre a primeira componente de sinal e seu último eco é uma fração significativa do período de símbolo do sinal transmitido, um símbolo transmitido pode chegar ao receptor durante a detecção do símbolo seguinte e causar interferência entre símbolos (*intersymbol interference* ou ISI) e prejudicar consideravelmente a detecção do símbolo transmitido. Em altas taxas de dados, o período de símbolo é curto, portanto, um pequeno atraso é suficiente para causar ISI.

Orthogonal frequency division multiplexing (OFDM) surgiu como uma alternativa ao CDMA para a obtenção de taxas de transmissão de dados elevadas. É utilizada no Wi-Fi, WiMAX e também no LTE. É uma técnica de multiplexação na frequência usada como modulação multiportadora.

A ideia central da modulação multiportadora é dividir a sequência de símbolos de alta taxa de transmissão (e portanto com período de símbolo curto) que se quer transmitir em várias sequências em paralelo, de baixa taxa de transmissão. Em seguida, cada sequência é usada para modular portadoras diferentes. Dividindo a sequência de símbolos original em várias sequências paralelas de menor taxa de transmissão, aumenta-se a duração do símbolo de cada sequência paralela, de tal modo que o atraso entre a primeira componente de sinal e o último eco é apenas uma pequena fração da duração do símbolo, reduzindo a ISI.

No OFDM, as subportadoras são selecionadas de tal modo que sejam todas ortogonais entre si, evitando assim a necessidade de se preocupar com interferência entre portadoras. Além disso, qualquer interferência residual entre símbolos pode ser eliminada pela utilização de intervalos de guarda entre os símbolos OFDM, cujo tempo de duração é feito maior do que o atraso entre a primeira componente de sinal e seu último eco. Ao fazer o intervalo de guarda maior do que o atraso esperado, a ISI pode ser completamente eliminada.

OFDM pode ser facilmente implementado usando a transformada rápida de Fourier cujas exigências computacionais são normalmente mais baixas do que a de um equalizador capaz de combater a ISI causada pelo multi-percurso. Além disso, é usado também no enlace direto do LTE como um esquema de múltiplo acesso. Chamado de OFDMA, consiste em dividir as subportadoras entre os usuários.

Apesar de possuir vários pontos positivos, o OFDM possui também alguns pontos negativos. Um dos mais importantes é o alto valor da razão entre a potência de pico e a potência média (PAPR, do inglês *peak-to-average power ratio*) do sinal OFDM. Isso faz com que o sinal seja distorcido após passar por um amplificador linear de radiofrequência. Resolver este problema requer o uso de amplificadores caros ou ineficientes energeticamente [2].

2.2.2 MIMO

O padrão LTE fornece amplo suporte ao uso de técnicas de MIMO para melhorar a robustez de enlace, a capacidade do sistema e a eficiência espectral. As técnicas MIMO usadas no LTE são:

Diversidade de transmissão: É usada para combater o desvanecimento do sinal causado pelo canal. A ideia central é enviar cópias do mesmo sinal, codificadas diferentemente, em várias antenas transmissoras. A diversidade de transmissão do LTE é baseada em técnicas de codificação SFBC (*space-frequency block coding*).

Beamforming: múltiplas antenas transmitem o mesmo sinal com amplitude e fase apropriadamente ponderadas de acordo com a posição do receptor de forma que se tenha um direcionamento da potência do sinal na direção do receptor. Além de enviar mais potência ao receptor, menos interferência é criada para outros usuários. Com efeito, melhora-se a razão sinal-interferência para todo o sistema.

Multiplexação espacial: A ideia por trás da multiplexação espacial é que múltiplos e independentes fluxos de dados podem ser transmitidos em paralelo usando-se múltiplas antenas de transmissão e separados no receptor usando múltiplas antenas de recepção, por meio de técnicas apropriadas de processamento de sinais. Isto pode ser feito desde que os diferentes canais formados entre as diferentes antenas de transmissão e recepção sejam suficientemente descorrelacionados, o que é verdade apenas em um ambiente muito dispersivo, em que o canal varia drasticamente no espaço. Em teoria, a multiplexação espacial pode conferir ganho na taxa de transmissão de dados proporcional ao número de antenas utilizado. O padrão LTE utiliza multiplexação espacial para até quatro antenas de transmissão e quatro antenas de recepção.

2.3 Canais do LTE

Assim como em outros padrões de telecomunicações, o LTE é organizado em uma pilha de protocolos e é composto pelas seguintes camadas, como mostrado na Figura 2.1:

Radio resource control (RRC), packet data convergence protocol (PDCP), radio link control

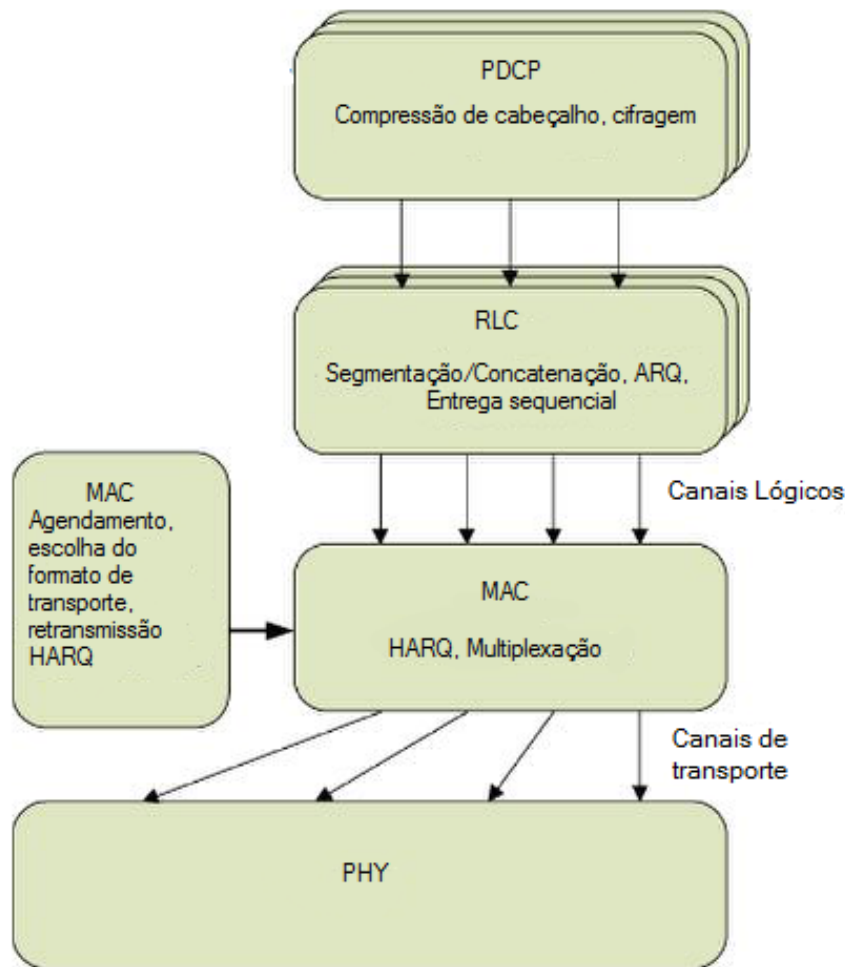


Figura 2.1: As camadas do sistema LTE [2].

(RLC), *medium access control* (MAC) e camada física (*physical layer* - PHY). A figura 2.1 mostra a camada 2, de acordo com o modelo OSI (*open systems interconnection*), do padrão LTE e suas sub-camadas, PDCP, RLC e MAC, e a camada 1 (camada física). A RRC pertence a camada 3, no modelo OSI, e por isso é omitida na figura. A seguir, serão descritas as funções da camada MAC e da camada PHY:

Camada MAC: algumas das principais funções da camada MAC são a correção de erros por meio de HARQ e o mapeamento entre canais lógicos e canais de transporte. A camada MAC é também responsável por determinar a escolha do formato de transmissão, o que inclui o tipo de modulação, taxa de codificação e modo MIMO.

Camada PHY: A principal função da camada física é a transmissão e recepção, de fato, de dados na forma de blocos de transporte. A camada física é também responsável pela sinalização de vários mecanismos de controle, como realimentação HARQ e medidas de canal.

Para viabilizar várias classes de serviços QoS (*quality of service*), o LTE adota uma estrutura hierárquica de canais. Diferentemente dos sistemas de terceira geração, o LTE possui somente canais compartilhados ou de *broadcast*, não há canais dedicados (em um sentido de comutação de

circuitos) para transmitir dados a usuários específicos. Existem três tipos diferentes de canais: os canais lógicos, os canais de transporte e os canais físicos. Como o foco do trabalho é o enlace direto, não serão abordados canais exclusivos do enlace de subida.

Os canais lógicos proveem serviços entre as camadas MAC e RLC. Cada canal lógico é definido baseado no tipo de informação que ele carrega. Existem muitos canais lógicos. Os mais importantes para este trabalho são os seguintes:

Broadcast control channel (BCCH): canal de enlace direto usado para fazer o *broadcast* de informações de controle do sistema aos usuários em uma célula. É mapeado em dois canais de transporte: o *Broadcast Channel* (BCH) e o *Downlink Shared Channel* (DL-SCH).

Common control channel: Canal bidirecional usado para transmitir informação de controle entre a rede e usuário quando o usuário ainda não faz parte da rede. O canal CCCH é usado durante o procedimento de acesso aleatório, realizado para conectar o usuário à rede.

Dedicated control channel (DCCH): Canal bidirecional que provê uma conexão dedicada e ponto-a-ponto entre o usuário e a rede. Usado para transmitir informações de controle.

Dedicated traffic channel (DTCH): Canal bidirecional que provê uma conexão ponto-a-ponto entre a rede e o usuário, utilizado para transmitir dados do usuário.

Os canais de transporte proveem serviços entre as camadas MAC e PHY. Um canal de transporte é caracterizado pela forma de transferência dos dados, isto é, qual é o código corretor de erros usado, o esquema de modulação e o mapeamento de antenas. O canal de transporte mais importante para este trabalho é o *Downlink Shared Channel* (DL-SCH).

Downlink shared channel (DL-SCH): É usado para transmitir dados no enlace direto, incluindo tanto dados de controle e dados do usuário. Possui suporte à HARQ, adaptação dinâmica de enlace e alocação dinâmica e semi-persistente de recursos. É o canal simulado neste trabalho.

Os dados nos canais de transporte são organizados em blocos de transporte. O tempo gasto na transmissão de um bloco de transporte é chamado de *transmission time interval* (TTI) e vale 1 ms no padrão LTE. Desconsiderando esquemas de multiplexação espacial, no máximo um bloco de transporte é transmitido a um usuário em cada TTI.

Os canais físicos são as implementações de fato dos canais de transporte. O canal físico relacionado ao canal DL-SCH é o *physical downlink shared channel* (PDSCH). Sua função é carregar o tráfego do usuário (não carrega informações de controle) no enlace direto.

Um canal físico corresponde a um conjunto de elementos de recurso na grade tempo-frequência. Um elemento de recurso corresponde ao uso de uma portadora durante um símbolo OFDM que carrega um símbolo modulado. Um bloco de recurso é um conjunto de elementos de recurso e representa, no domínio da frequência, a quantidade mínima de recursos da rede que podem ser alocados.

A Figura 2.2 mostra o mapeamento dos canais lógicos, de transporte e físicos. Todos os canais lógicos citados anteriormente são mapeados no DL-SCH.

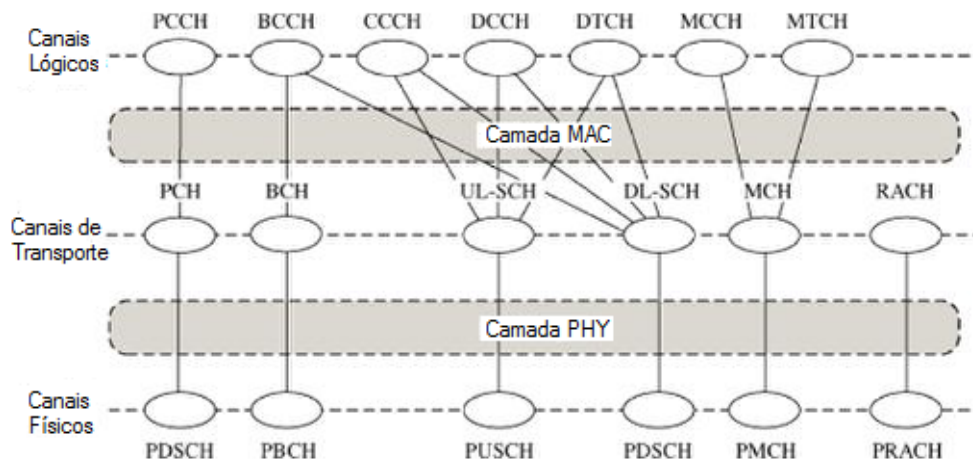


Figura 2.2: Mapeamento entre diferentes tipos de canais [2].

2.4 Estrutura do Frame LTE e Grade de Recursos

O *frame* LTE descreve a forma básica de organização dos recursos da rede no domínio do tempo. A transmissão no enlace direto é baseada no símbolo OFDM com prefixo cíclico e será analisada nesta seção.

O padrão LTE possui dois modos de duplexação: FDD (*frequency division duplexing*) e TDD (*time division duplexing*). A estrutura do *frame* muda dependendo do tipo de duplexação. Neste trabalho será analisado apenas o *frame* FDD.

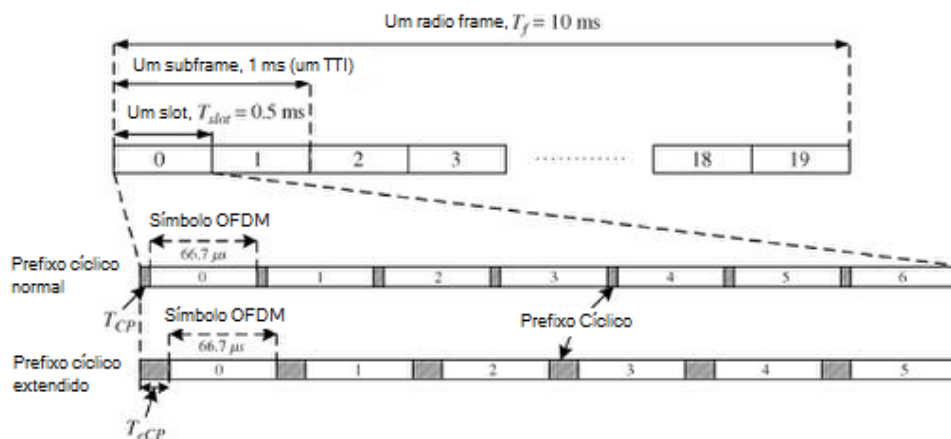


Figura 2.3: *Radio frame* do LTE [2].

A menor unidade do *frame* é o *slot* que tem tamanho igual a 0,5 ms. Dois *slots* consecutivos definem um *subframe*, que tem duração de 1ms, e 20 *slots* constituem um *radio frame*, de duração igual a 10 ms. Logo, a duração de um *subframe* corresponde ao TTI mínimo no enlace direto, que vale 1 ms. Cada *slot* consiste em um número de símbolos OFDM incluindo prefixos cíclicos. Como pode ser visto na Figura 2.3, cada *slot* contém 7 símbolos OFDM se for utilizado o prefixo cíclico de tamanho normal ou 6 símbolos OFDM se for utilizado o prefixo cíclico estendido. Para

espaçamento entre subportadoras igual a $\Delta f = 15kHz$, a duração de um símbolo OFDM é $\frac{1}{\Delta f} = 66.7\mu s$ como mostra a Figura 2.3.

O recursos físicos da rede no enlace direto são descritos por uma grade tempo-frequência chamada grade de recursos, ilustrada na Figura 2.4. A representação dos recursos da rede em um plano tempo-frequência torna intuitivo o entendimento da alocação de recursos. Cada coluna e cada linha da grade de recursos correspondem a um símbolo OFDM e a uma portadora OFDM, respectivamente. A duração da grade de recursos no domínio do tempo corresponde a um *slot* no *radio frame*. A menor unidade de tempo-frequência na grade de recursos é chamada de elemento de recurso. A grade de recurso consiste em um conjunto de blocos de recurso, que por sua vez são formados por um conjunto de elementos de recurso.

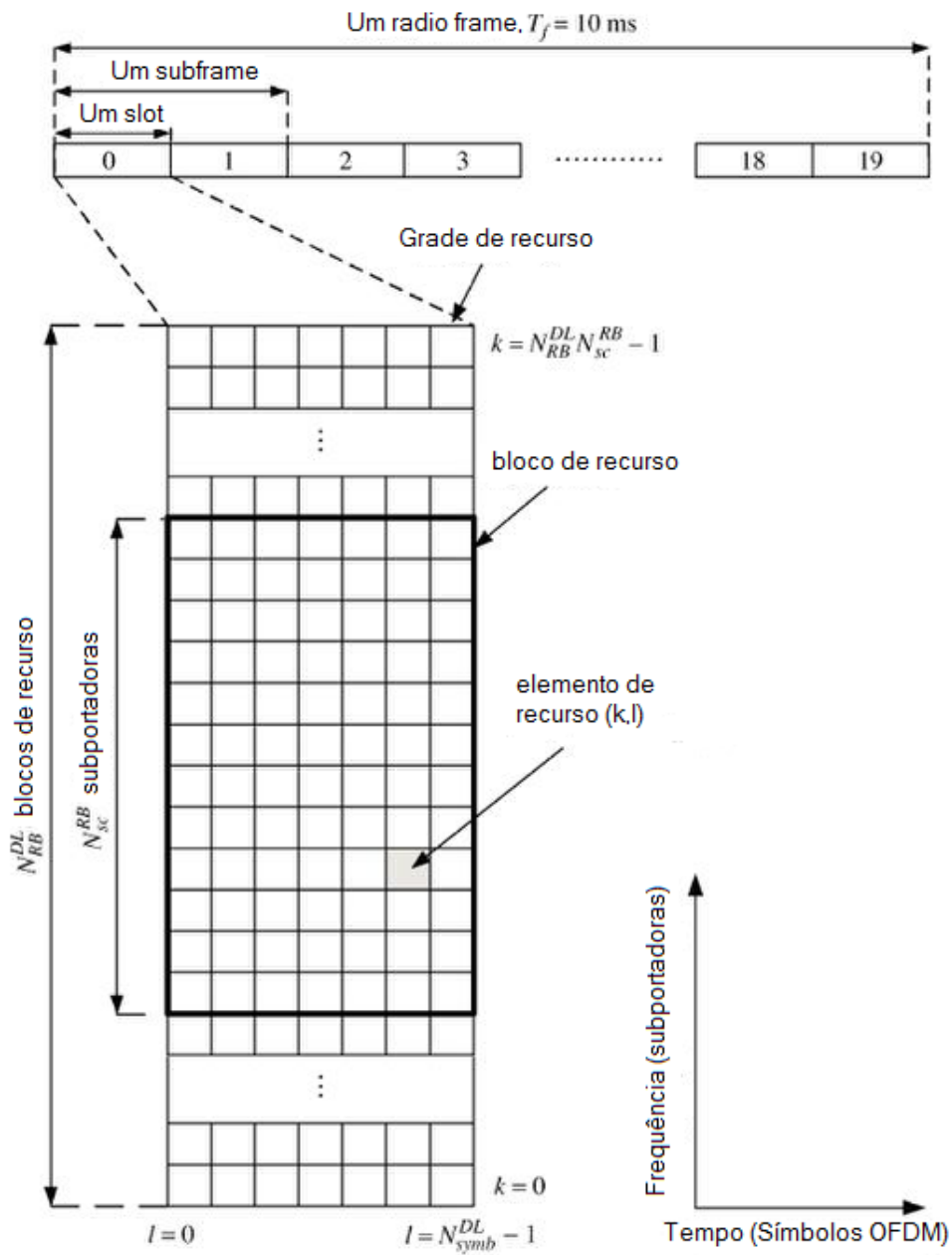


Figura 2.4: Grade de recurso [2].

Capítulo 3

CRC e Códigos Turbo

Neste capítulo é exposto o funcionamento dos códigos detectores e corretores de erros usados no canal DL-SCH do padrão LTE.

3.1 CRC

CRC (*cyclic redundancy check*) é um código detector de erros usado em sistemas de comunicação e em dispositivos de armazenamento de dados com o objetivo de detectar erros em sequências de *bits*. É importante notar que a CRC não corrige erros e sim, apenas sinaliza que em dada sequência de *bits* há erro(s).

Descreve-se a seguir o processo de codificação CRC. A partir dos *bits* da mensagem forma-se um polinômio tomando cada *bit* como um coeficiente. Por exemplo, para a mensagem $m = [111]$, o polinômio correspondente é $m(D) = D^2 + D + 1$. Se $m = [1011]$, $m(D) = D^3 + D + 1$, e assim por diante. A codificação CRC é baseada na divisão euclidiana de polinômios:

$$D(D) = G(D)Q(D) + R(D).$$

O processo consiste em dividir o dividendo $D(D) = D^n m(D)$ por um polinômio $G(D)$ chamado polinômio gerador em que n é o grau do polinômio gerador. O resto da divisão $R(D)$ corresponde aos *bits* de paridade CRC e os *bits* correspondentes são anexados ao final da mensagem formando a palavra - código. O quociente $Q(D)$ não é utilizado.

O receptor da mensagem sabe qual polinômio gerador foi utilizado e portanto conhece n . O receptor então separa os *bits* de paridade transmitidos dos *bits* de informação, forma o polinômio $\hat{m}(D)$ e repete a divisão feita no transmissor:

$$\hat{D}(D) = G(D)\hat{Q}(D) + \hat{R}(D).$$

Se $\hat{R}(D) = R(D)$ não há erros na mensagem. Caso contrário, há erros e a mensagem deve ser retransmitida.

Uma CRC de n *bits* insere n *bits* de redundância ao fim de uma mensagem de tamanho arbitrário. A CRC de n *bits* detecta qualquer sequência de erros em rajada (sequência de erros contíguos) de tamanho igual ou menor a n e detecta uma fração igual à $1 - 2^{-n}$ de todas as

seqüências de erros em rajada de tamanho maior que n [3].

O cálculo dos *bits* de CRC é simples e pode ser feito com um registrador de deslocamentos com realimentação linear. Sua simplicidade, aliada ao bom desempenho detector de erros ajuda a explicar sua adoção no padrão LTE e em outros sistemas de comunicação [3].

No padrão LTE são utilizados os seguintes polinômios geradores [5]:

- $g_{CRC24A}(D) = [D^{24} + D^{23} + D^{18} + D^{17} + D^{14} + D^{11} + D^{10} + D^7 + D^6 + D^5 + D^4 + D^3 + D + 1]$
para CRC de tamanho $L = 24$.

- $g_{CRC24B}(D) = [D^{24} + D^{23} + D^6 + D^5 + D + 1]$ para CRC de tamanho $L = 24$.

- $g_{CRC16}(D) = [D^{16} + D^{12} + D^5 + 1]$ para CRC de tamanho $L = 16$.

- $g_{CRC8}(D) = [D^8 + D^7 + D^4 + D^3 + D + 1]$ para CRC de tamanho $L = 8$.

O foco deste trabalho é no canal DL-SCH e, no caso específico do bloco de transporte no canal DL-SCH, o polinômio usado é $g_{CRC24A}(D)$. Caso haja segmentação de blocos, em cada bloco segmentado é aplicada nova CRC utilizando o polinômio $g_{CRC24B}(D)$.

3.2 Códigos Turbo

3.2.1 Introdução

Códigos Turbo são códigos convolucionais paralelamente concatenados com entrelaçamento (*parallel concatenated convolutional code with interleaving*) e foram inventados na França por Berrou no início dos anos 1990 [6]. Foram os primeiros códigos práticos a se aproximarem da capacidade de canal de um canal do tipo AWGN (*additive white gaussian noise*) [7]. Foram utilizados na terceira geração de sistemas de comunicação (3G) e são utilizados no sistema LTE e em sistemas de comunicação via satélite. Juntamente com os códigos LDPC (*low density parity codes*), são os códigos de melhor desempenho existentes atualmente.

Os códigos turbo são feitos a partir de códigos componentes mais simples paralelamente concatenados. A seqüência de informação de entrada de cada codificador em paralelo é uma versão entrelaçada da seqüência original. O grande diferencial dos códigos turbo está em seu processo de decodificação iterativo. Nesta seção serão explicados de forma sucinta os fundamentos da codificação turbo.

3.2.2 Códigos Convolucionais

O objetivo de um código corretor de erros é melhorar a confiabilidade de um canal de comunicação inserindo *bits* redundantes à mensagem. Para entender como os códigos convolucionais funcionam, é útil relembrar algumas características dos códigos de bloco.

O processo de codificação de um código de bloco é descrito a seguir. Suponha que deseje-se codificar e posteriormente transmitir uma seqüência de L *bits*. Primeiramente, os L *bits* são

divididos em blocos de k bits, pois o processo de codificação é feito por blocos de k bits. A cada etapa, um bloco de k bits é transformado em um bloco de n bits pela matriz geradora do código de bloco, e ao final do processo, os blocos de n bits são concatenados, formando a sequência de saída do codificador.

A palavra-código produzida pelo codificador em um código de bloco depende somente da sequência de bits de entrada. Há uma correspondência direta e única entre a mensagem $u = [u_1 u_2 \dots u_k]$ e sua palavra-código $x = [x_1 x_2 \dots x_n]$. O processo de decodificação também é feito por blocos. O trabalho do decodificador consiste em encontrar a palavra-código pertencente ao conjunto de palavras-código possíveis que possui a menor distância (de Hamming ou euclidiana) da palavra-código recebida.

O processo de codificação em um código convolucional é feito de maneira diferente. A codificação é feita continuamente e não por blocos. Pode-se pensar que todos os L bits da sequência de entrada geram uma grande palavra-código de uma só vez. O codificador é uma máquina de estados e, portanto, possui memória, em oposição ao código de blocos. O código convolucional é, portanto, melhor representado por um diagrama de estados ou um diagrama de treliça em detrimento da representação por matriz geradora. O trabalho do decodificador consiste em encontrar a sequência mais provável de estados baseado na sequência de bits recebida.

Define-se a taxa de codificação R como a razão entre a quantidade de bits na entrada do codificador e a quantidade de bits na saída do codificador, e é dada por:

$$R = \frac{k}{n}.$$

A seguir, descreve-se o processo de codificação convolucional.

3.2.2.1 Codificador

O codificador de um código convolucional é uma máquina de estados e pode ser implementado na prática por um registrador de deslocamento. A Figura 3.1 mostra um codificador convolucional.

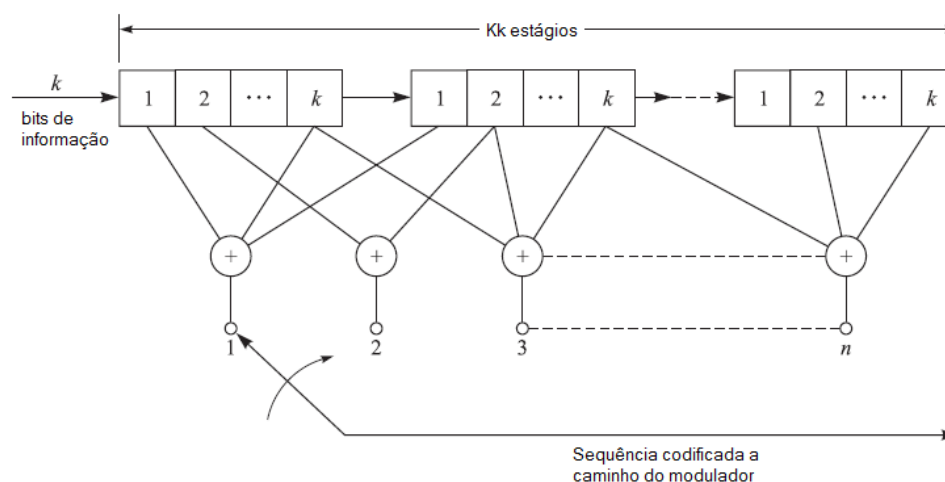


Figura 3.1: Codificador convolucional [6].

Há três parâmetros que descrevem o código convolucional, k , n e K . K é chamado de comprimento de *bits* limitantes (*constraint length*) e é o número de estágios do registrador de deslocamentos, k é o número de *bits* contidos em cada estágio e n é o número de *bits* gerados a cada etapa do processo de codificação. A cada etapa da codificação, k *bits* de informação são inseridos dentro do primeiro estágio do registrador de deslocamentos, que possui no total K estágios com k *bits* por estágio. Simultaneamente, os k *bits* que ocupavam o primeiro estágio são jogados para o segundo estágio. Os *bits* do segundo estágio são jogados para o terceiro estágio e assim por diante. Os *bits* do último estágio são descartados. Os n *bits* de saída do codificador são gerados pela operação OU-EXCLUSIVO (soma módulo-2) entre certas casas do registrador de deslocamento. A taxa de codificação é $R = k/n$ [6].

Utilizam-se vetores para representar a maneira na qual estão ligadas as casas responsáveis por gerar os *bits* de saída. Tem-se n vetores no total, um para cada *bit* gerado pelo codificador. Cada vetor tem tamanho Kk . Se a i -ésima casa estiver envolvida na operação OU-EXCLUSIVO para o j -ésimo *bit* de saída, o i -ésimo *bit* do vetor j é 1. Caso contrário o i -ésimo *bit* é 0. Apresenta-se a seguir um exemplo adaptado de [6].

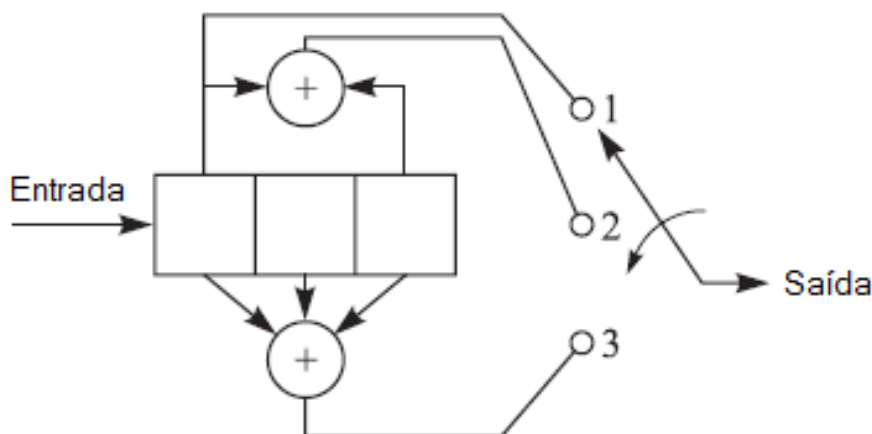


Figura 3.2: Codificador convolucional $k = 1$, $K = 3$ e $n = 3$ [6].

Seja o codificador convolucional da Figura 2.6 com $k = 1$, $K = 3$ e $n = 3$. A taxa do código é, portanto, $1/3$. Inicialmente, considera-se que todas as casas do registrador de deslocamentos são iguais a zero. O *bit* de saída 1 é simplesmente igual à primeira casa, porque ele está diretamente ligado à primeira casa. O *bit* de saída 2 é gerado pelo OU-EXCLUSIVO do *bit* da primeira casa do registrador de deslocamentos e da última casa. O *bit* de saída 3 é gerado pelo OU-EXCLUSIVO dos *bits* das 3 casas. Suponha que o primeiro *bit* de entrada é 1. Então a sequência de saída será 111. Se o segundo *bit* for 0, a sequência de saída será 001. Se o terceiro *bit* for 1, a sequência de saída será 100 e assim por diante. Os 3 vetores que descrevem a forma na qual estão ligadas as casas responsáveis por gerar os *bits* de saída são:

$$\mathbf{g}_1 = [100] \text{ para o } \textit{bit} \text{ de saída } 1,$$

$\mathbf{g}_2 = [101]$ para o *bit* de saída 2,

$\mathbf{g}_3 = [111]$ para o *bit* de saída 3.

Os vetores \mathbf{g}_1 , \mathbf{g}_2 e \mathbf{g}_3 são a resposta impulsional entre a sequência de entrada \mathbf{u} e as três saídas \mathbf{c} , que são dadas por:

$$\mathbf{c}^{(1)} = \mathbf{u} * \mathbf{g}_1,$$

$$\mathbf{c}^{(2)} = \mathbf{u} * \mathbf{g}_2,$$

$$\mathbf{c}^{(3)} = \mathbf{u} * \mathbf{g}_3.$$

em que $*$ indica a operação de convolução no corpo de Galois $\text{GF}(2)$. A sequência de saída \mathbf{c} para este exemplo é o resultado do entrelaçamento entre $\mathbf{c}^{(1)}$, $\mathbf{c}^{(2)}$ e $\mathbf{c}^{(3)}$:

$$\mathbf{c} = (c_1^{(1)}, c_1^{(2)}, c_1^{(3)}, c_2^{(1)}, c_2^{(2)}, c_2^{(3)}, \dots).$$

Sabendo que a operação de convolução é equivalente à multiplicação no domínio da transformada Z , define-se a transformada D de \mathbf{u} como:

$$u(D) = \sum_{i=0}^{\infty} u_i D^i$$

Nota-se que a transformada D é igual à transformada Z substituindo-se z^{-1} por D . As três funções de transferências relativas às três respostas impulsionalis \mathbf{g}_1 , \mathbf{g}_2 e \mathbf{g}_3 são:

$$g_1(D) = 1,$$

$$g_2(D) = 1 + D^2,$$

$$g_3(D) = 1 + D + D^2.$$

A transformadas das saídas são:

$$c^{(1)}(D) = u(D)g_1(D),$$

$$c^{(2)}(D) = u(D)g_2(D),$$

$$c^{(3)}(D) = u(D)g_3(D).$$

Em forma matricial tem-se:

$$\mathbf{c}(\mathbf{D}) = \mathbf{u}(\mathbf{D})\mathbf{G}(\mathbf{D}),$$

$$\text{em que } \mathbf{G}(\mathbf{D}) = [g_1(D)g_2(D)g_3(D)].$$

Como neste exemplo as três sequências de saída são entrelaçadas, a transformada da saída do codificador é:

$$c(D) = c^{(1)}(D^3) + Dc^{(2)}(D^3) + D^2c^{(3)}(D^3).$$

A partir deste exemplo, pode-se generalizar a idéia para o caso geral em que K , k e n são valores quaisquer.

Foi visto que para implementar o código, utiliza-se um registrador de deslocamento de K estágios com k bits em cada estágio. Entretanto, de modo geral, em vez de utilizar um único registrador de deslocamentos de K estágios e com k bits por estágio, é possível implementar o

mesmo código convolucional utilizando k registradores de deslocamento de K estágios com apenas 1 *bit* por estágio. Dessa forma, a implementação de um código com $k > 1$ pode ser reduzida à uma versão com múltiplas entradas e múltiplas saídas do exemplo anterior[6]. A Figura 3.3 mostra dois codificadores que implementam o mesmo código.

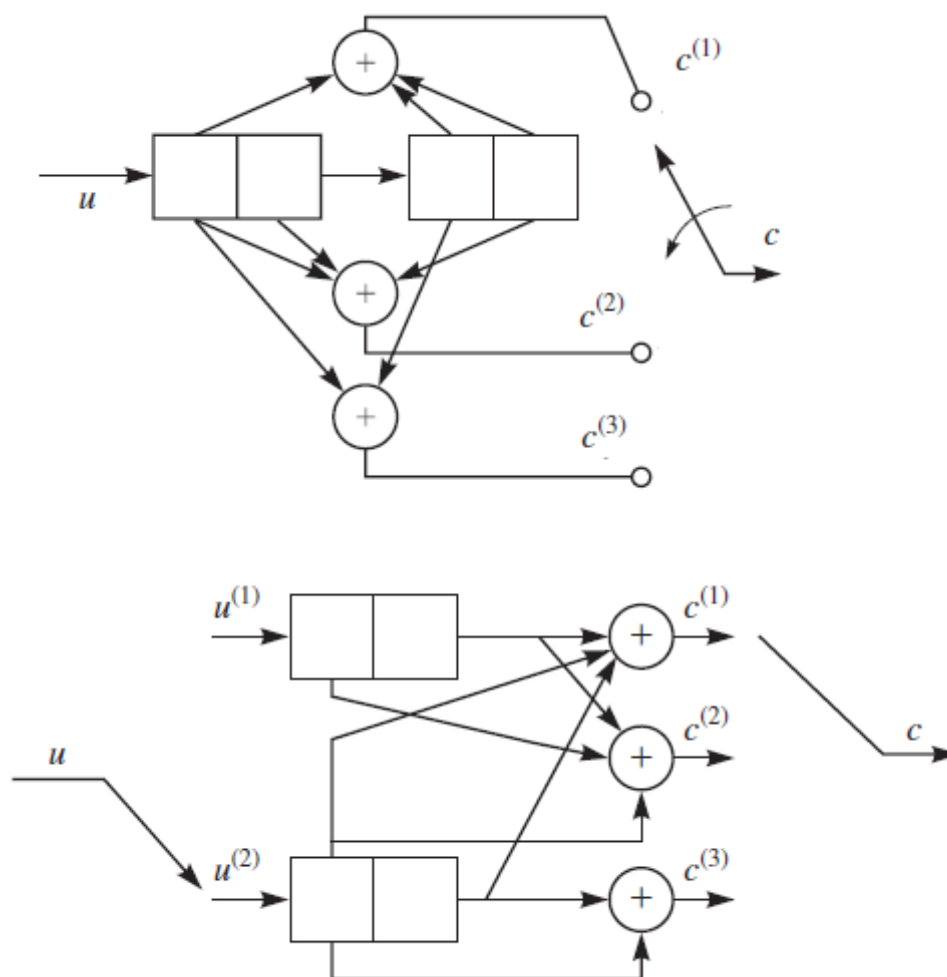


Figura 3.3: Codificadores Equivalentes [6].

Para o caso geral, tem-se então k registradores de deslocamento, e cada um dos k registradores se relaciona com cada saída n através de uma função de transferência. Com efeito, para o caso geral, a matriz \mathbf{G} é uma matriz $k \times n$ chamada matriz geradora no domínio da transformada. Cada um de seus elementos g_{ij} é a função de transferência entre a entrada i e a saída j . A matriz geradora no domínio da transformada descreve totalmente o código e é o análogo do código convolucional à matriz geradora dos códigos de bloco.

3.2.2.2 Treliça e Diagrama de Estados

É sabido que, conhecendo-se a matriz geradora, é possível calcular a sequência de *bits* codificados, porém alternativamente representa-se um código convolucional por meio de seu diagrama

de estados ou treliça.

Os estados do codificador são as diferentes configurações de *bits* possíveis em sua memória. A memória é representada pelas K casas do registrador de deslocamento. Como a primeira casa é sempre usada para armazenar os k *bits* de entrada a cada etapa do processo de codificação, considera-se que a memória do codificador consiste nas $K - 1$ casas restantes. Cada casa armazena k *bits* e cada *bit* pode assumir dois valores, portanto, um código convolucional com *constraint length* K possui $2^{k(K-1)}$ diferentes configurações de sua memória (ou estados). O próximo estado e os *bits* de saída são determinados pelos *bits* de entrada e pelo estado atual.

O diagrama de estados é uma representação compacta que ajuda muito a visualizar o funcionamento do código convolucional. A Figura 3.4 mostra o diagrama de estados para o codificador da figura 3.1. No diagrama, os quatro estados (a,b,c e d) são representados por retângulos. As setas representam as transições de estado. Setas contínuas representam a transição de estado se o *bit* de entrada for igual a 0. Já as setas pontilhadas representam a transição de estado se o *bit* de entrada for 1. Os 3 *bits* de saída são gerados na transição de estado e, portanto, acompanham as setas.

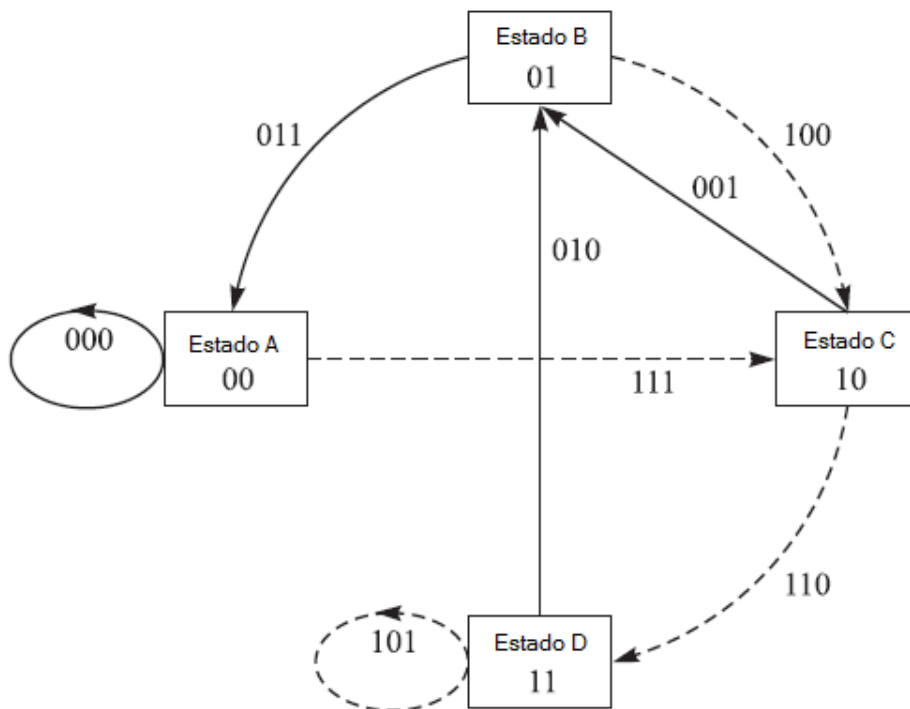


Figura 3.4: Diagrama de Estados [6].

No diagrama de treliça, (Figura 3.5) cada ponto representa um dos possíveis estados do codificador e cada traço representa uma transição de estado. Assim como no diagrama de estados, traços contínuos representam a transição de estado se o *bit* de entrada for igual a zero e traços pontilhados representam a transição de estado se o *bit* de entrada for igual a um. Acompanhando os traços de transição, tem-se os *bits* de saídas gerados. Após algumas etapas, o codificador atinge o estado estacionário. A partir deste ponto, o diagrama de treliça passa a se repetir a cada $K - 1$

etapas.

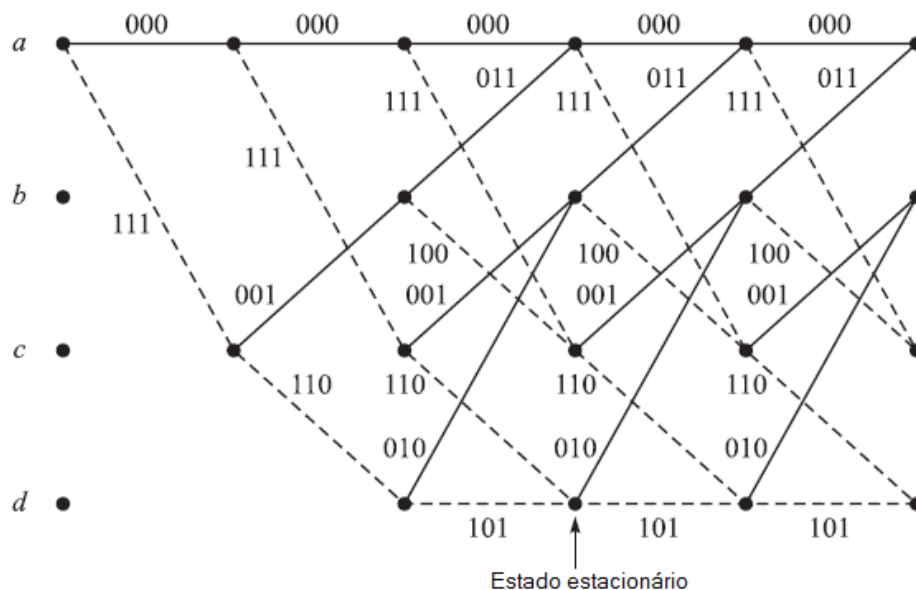


Figura 3.5: Diagrama de Treliça [6].

O foco do diagrama de estados é destacar os estados do codificador. Já o diagrama de treliça enfatiza os possíveis caminhos percorridos pelo código a cada etapa. As duas representações são alternativas gráficas úteis à representação por matriz geradora.

3.2.2.3 Códigos Sistemáticos e Recursivos

Um codificador sistemático é um codificador no qual a sequência de informação (sequência de entrada do codificador) aparece diretamente em sua saída como parte do código [6]. Por exemplo, o codificador da Figura 3.2 é um codificador sistemático, uma vez que:

$$\mathbf{c}^{(1)} = \mathbf{u} * \mathbf{g}_1 = \mathbf{u}.$$

Um codificador recursivo é um codificador cujas saídas são retroalimentadas à entrada. A matriz geradora \mathbf{G} , discutida anteriormente é formada por razões de polinômios, ao contrário dos códigos não-recursivos, que possuem matrizes geradoras cujos elementos são apenas polinômios. Consequentemente, os codificadores recursivos possuem resposta impulsional infinita, diferentemente dos codificadores não-recursivos, que possuem resposta impulsional finita. O codificador da Figura 3.2 é não-recursivo e o da Figura 3.6 é recursivo.

3.2.3 Códigos Concatenados e codificador Turbo

Um esquema de codificação concatenado foi proposto primeiramente por Forney e consiste em combinar dois ou mais códigos com o objetivo de aumentar os ganhos de codificação. Utilizando códigos simples como componentes do código concatenado, é possível obter uma capacidade cor-

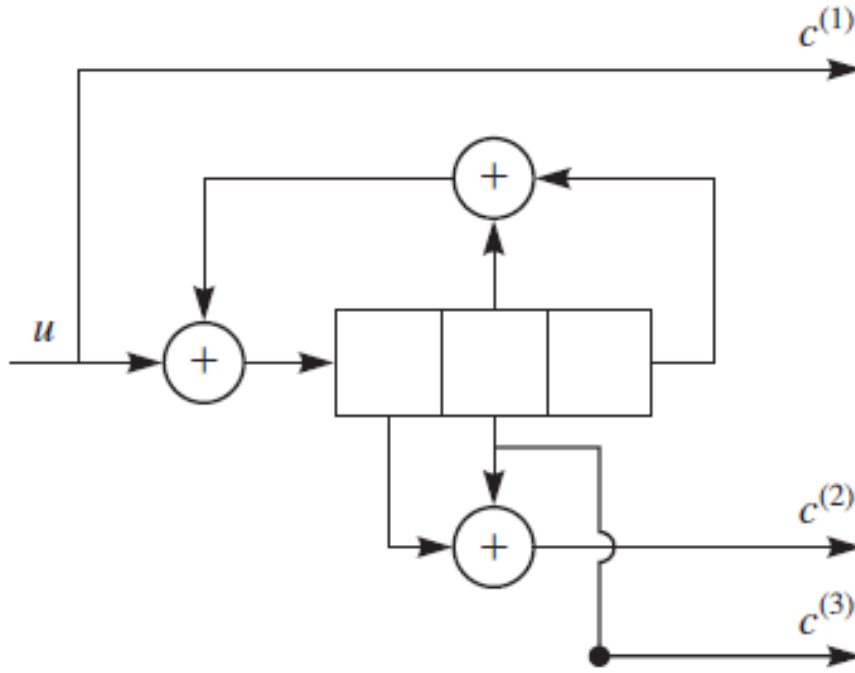


Figura 3.6: Codificador Sistemático Recursivo [6].

retora de erros comparável à códigos muito maiores, porém sem precisar utilizar codificadores e decodificadores excessivamente complexos [8].

Códigos componentes podem ser concatenados de maneira serial ou paralela. Na concatenação serial, os códigos são concatenados em sequência, isto é, a sequência de entrada alimenta o primeiro codificador, em seguida a sequência de saída do primeiro codificador passa por um entrelaçador e, após entrelaçamento, alimenta o segundo codificador. Na concatenação paralela os codificadores são alimentados pela mesma sequência de entrada, entrelaçada de forma diferente para cada codificador, e cada codificador gera uma sequência de *bits* codificados independente dos outros codificadores.

Os códigos turbo são códigos convolucionais paralelamente concatenados. O codificador turbo é composto por dois codificadores convolucionais sistemáticos e recursivos (*Recursive Systematic Convolutional (RSC)* com taxa 1/2). A sequência de *bits* de informação \mathbf{u} é alimentada diretamente ao primeiro codificador. Para o outro codificador a sequência \mathbf{u} sofre um processo de entrelaçamento, em que seus *bits* são permutados, e uma nova sequência $\mathbf{u}^{(i)}$ é criada. O processo de entrelaçamento altera a ordem dos *bits* da sequência para o segundo codificador e, como resultado, tem-se 2 sequências $\mathbf{u}^{(i)}$. A sequência $\mathbf{u}^{(i)}$ é então alimentada ao i -ésimo codificador. Para manter a coerência na notação empregada, faz-se $\mathbf{u}^{(1)} = \mathbf{u}$. Cada codificador i produz uma sequência de *bits* sistemáticos igual à $\mathbf{u}^{(i)}$ e uma sequência de *bits* de paridade $\mathbf{v}^{(i)}$. A sequência $\mathbf{u}^{(2)}$ é descartada e os *bits* de saída do codificador são formados pelas sequências restantes: a sequência sistemática $\mathbf{u}^{(1)} = \mathbf{u}$ e as sequências de paridade $\mathbf{v}^{(i)}$, $i = 1, 2$.

No padrão LTE, os códigos turbo são utilizados no canal DL-SCH. Os dois códigos componentes

usados são iguais, têm 8 estados e possuem a seguinte matriz geradora [5]:

$$\mathbf{G} = \left[1, \frac{g_1(D)}{g_0(D)} \right],$$

em que $g_0(D) = 1 + D^2 + D^3$ e $g_1(D) = 1 + D + D^3$.

A Figura 3.7 mostra o diagrama do codificador utilizado no canal DL-SCH.

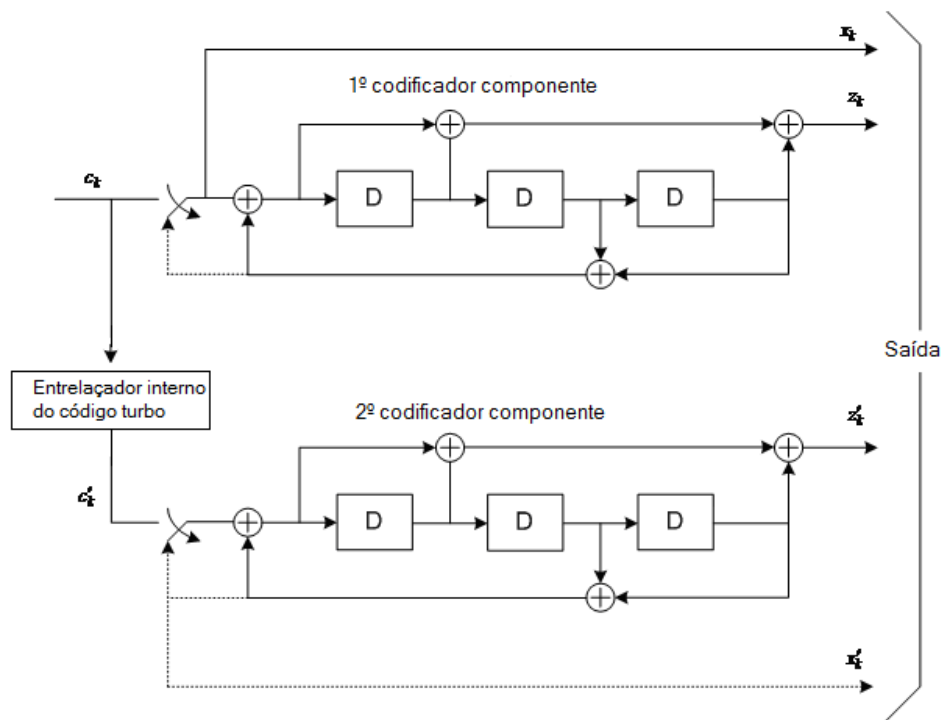


Figura 3.7: Codificador Turbo LTE [5].

3.2.4 Decisão rígida e decisão suave (*hard-decision* e *soft-decision*)

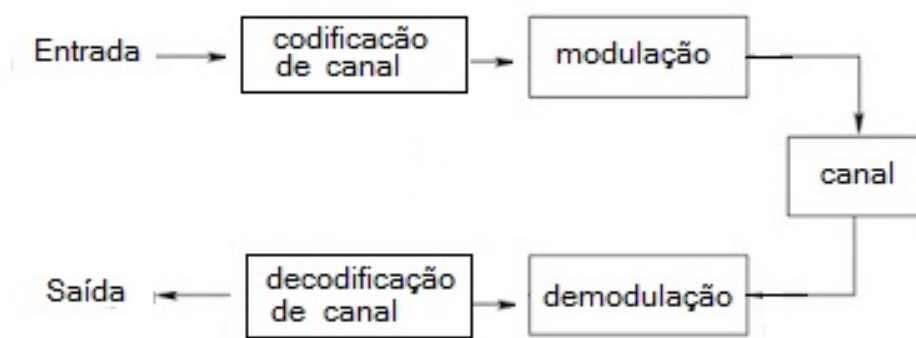


Figura 3.8: Sistema de comunicação [9].

No sistema de comunicação representado na Figura 3.8, a função do demodulador é, a partir do

sinal $s_r[k] = s_t[k] + n[k]$ recebido, corrompido por ruído (suposto aditivo gaussiano branco), estimar quais símbolos foram enviados. A decisão é feita calculando-se as probabilidades *a posteriori* (*A Posteriori Probability* ou APP):

$$P(s_t[k] = i | s_r[k]) \forall i \in A,$$

em que A é o conjunto de M símbolos pertencentes ao alfabeto do esquema de modulação digital empregado. O símbolo mais provável de ter sido enviado é aquele cuja probabilidade *a posteriori* é maior. Pelo teorema de Bayes:

$$P(s_t[k] = i | s_r[k]) = \frac{p(s_r[k] | s_t[k] = i) P(s_t[k] = i)}{p(s_r[k])},$$

$$p(s_r[k]) = \sum_{i=0} p(s_r[k] | s_t[k] = i) P(s_t[k] = i).$$

$P(s_t[k] = i)$ é chamada de probabilidade *a priori* e é a probabilidade de transmissão do símbolo i . O cálculo da probabilidade *a posteriori* pode ser pensado como um refinamento da probabilidade *a priori* graças à informação contida em $s_r[k]$. Se todos os símbolos possuem igual probabilidade de serem transmitidos, para tomar a decisão com a menor probabilidade de erro, o demodulador deve comparar as probabilidades condicionais $p(s_r[k] | s_t[k] = i)$, também chamadas de funções de verossimilhança (*likelihood functions*), e escolher a maior [9].

Chama-se estimação de máxima verossimilhança (*maximum likelihood estimation*) o método de demodulação descrito no parágrafo anterior. Normalmente, o processo de demodulação termina com o demodulador passando uma decisão rígida (*hard decision*), isto é, uma decisão final sobre o valor de um símbolo recebido após passagem por um canal ruidoso. Entretanto, a decodificação turbo trabalha com decisões suaves (*soft-decisions*) e não com decisões rígidas e portanto o demodulador deve operar de maneira diferente. Para exemplificar, considera-se um sistema de modulação BPSK (*Binary Phase Shift Keying*) em que o alfabeto da modulação possui apenas 2 símbolos e um canal AWGN. Para simplificar, o modulador neste exemplo mapeia os *bits* 0 e 1 nos símbolos -1 e 1.

O canal AWGN faz com que o sinal que chega ao demodulador seja $s_r[k] = s_t[k] + n[k]$, em que $s_t[k] \in \{-1, 1\}$ e $n[k]$ é uma realização de um processo gaussiano de média igual a 0 e variância igual a σ^2 e, portanto, é um número real. O demodulador deve a partir do sinal recebido $s_r[k]$ estimar o símbolo transmitido $s_t[k]$. A estimação de $s_t[k]$ pelo demodulador será chamada de $\hat{s}_t[k]$. Se o demodulador for do tipo SIHO (*soft Input/hard Output*) $\hat{s}_t[k]$ deve obrigatoriamente pertencer ao alfabeto do esquema de modulação em questão. No caso da modulação BPSK os símbolos do alfabeto são -1 e 1. Para este caso a estimação de verossimilhança é equivalente à avaliar o sinal de $s_r(k)$. Se o sinal de $s_r(k)$ for positivo, $\hat{s}_t(k) = 1$. Se o sinal de $s_r(k)$ for negativo, $\hat{s}_t(k) = -1$. De modo geral, o demodulador SIHO executa uma função matemática cujo domínio é o conjunto dos números reais e a imagem é o alfabeto do esquema de demodulação, isto é, ele mapeia a entrada $s_r(k)$ em apenas dois valores: -1 ou 1.

Introduz-se agora o conceito de decisão suave (*soft-decision*), muito importante para um decodificador turbo. Em vez de mapear o sinal recebido $s_r(k)$ em apenas dois valores, o demodulador poderia enviar mais informação a respeito do símbolo recebido. A ideia é enviar, além da decisão

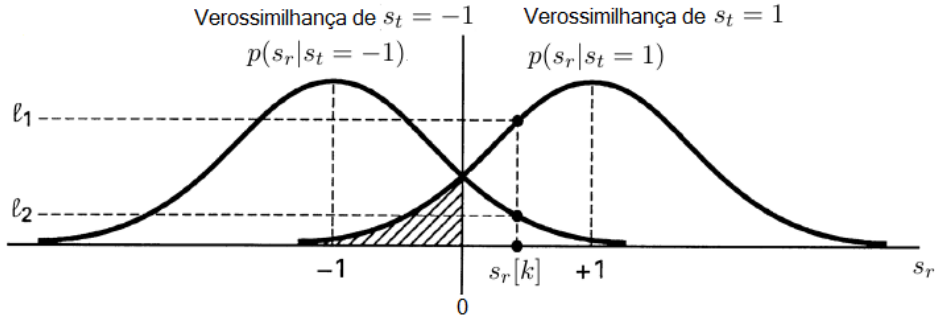


Figura 3.9: Funções de verossimilhança [9].

rígida (qual símbolo foi enviado), o grau de certeza sobre a decisão tomada. Em vez de dizer "s_t[k] é igual a 1" ou "s_t[k] é igual a 0" o demodulador SISO (*soft input/soft output*) diz "tenho quase certeza que s_t[k] é igual a 1", "penso que s_t[k] é igual a 1", "acho que s_t[k] é igual a 1, mas com pouca certeza". A função matemática operada pelo demodulador mapeia s_r(k) em um conjunto imagem cujo número de elementos é maior que 2, podendo ser até mesmo infinito no caso de uma demodulação analógica [9].

3.2.5 LLR

A decisão suave deve oferecer uma medida do grau de certeza sobre a decisão. Para gerar esta grandeza, utilizam-se novamente as probabilidades *a posteriori* citadas na sub-seção anterior. Para o exemplo da modulação BPSK, uma medida utilizada é a razão entre as duas probabilidades *a posteriori*, chamada de teste da razão de verossimilhança (*likelihood ratio test - LRT*):

$$LRT = \frac{P(s_t[k] = 1 | s_r[k])}{P(s_t[k] = -1 | s_r[k])}.$$

A fim de obter uma notação mais simplificada, de agora em diante será omitido o índice *k* da notação empregada para s_t[k], ŝ_t[k] e s_r[k].

Usando o teorema de Bayes, tem-se:

$$LRT = \frac{p(s_r | s_t = 1)P(s_t = 1)}{p(s_r | s_t = -1)P(s_t = -1)}.$$

Note que a decisão rígida está implícita na razão entre as duas probabilidades *a posteriori*. Se LRT for maior que 1, ŝ_t = 1 se LRT for menor que 1, ŝ_t = -1. Além disso, quanto mais próximo de 0, maior a probabilidade de ŝ_t = -1. Por outro lado quanto mais próximo de +∞, maior a probabilidade de ŝ_t = 1.

Alternativamente, pode-se aplicar o logaritmo à LRT e obter uma nova medida chamada de logaritmo da razão de verossimilhança (*log likelihood ratio* ou LLR):

$$L(s_t | s_r) = \log \left[\frac{P(s_t = 1 | s_r)}{P(s_t = -1 | s_r)} \right] = \log \left[\frac{p(s_r | s_t = 1)P(s_t = 1)}{p(s_r | s_t = -1)P(s_t = -1)} \right]$$

$$L(s_t | s_r) = \log \left[\frac{p(s_r | s_t = 1)}{p(s_r | s_t = -1)} \right] + \log \left[\frac{P(s_t = 1)}{P(s_t = -1)} \right]$$

$$L(s_t|s_r) = L(s_r|s_t) + L(s_t),$$

em que $L(s_r|s_t)$ é a LLR relativa à estimação do símbolo pelo demodulador após corrupção pelo ruído do canal e $L(s_t)$ é a LLR *a priori* do símbolo s_t .

Para simplificar a notação, reescreve-se a LLR como:

$$L'(\hat{s}_t) = L_c(s_r) + L(s_t),$$

em que $L'(\hat{s}_t) = L(s_t|s_r)$ e $L_c(s_r) = L(s_r|s_t)$. Esta notação enfatiza que $L_c(s_r)$ é a melhoria adicionada à LLR *a priori* do símbolo s_t , $L(s_t)$, proveniente da estimação do símbolo s_t feita pelo demodulador.

A LLR é um número real que representa a decisão suave do demodulador. O sinal da LLR dá a decisão rígida (1 ou -1). Quanto mais positiva for a LLR, maior a certeza de que o símbolo enviado é 1 e, logo, que o *bit* enviado é 1. Por outro lado, quanto mais negativa a LLR, maior a certeza de que o símbolo enviado é -1 e, logo, que o *bit* enviado é 0.

3.2.6 Decodificação iterativa

Após demodulação, as decisões suaves (LLRs) tomadas pelo demodulador são enviadas ao decodificador. No processo de decodificação iterativa, empregado nos códigos turbo, dois ou mais decodificadores são usados, um para cada codificador convolucional utilizado no processo de codificação. No sistema LTE, 2 codificadores convolucionais são usados e portanto 2 decodificadores são utilizados na decodificação. Os decodificadores são do tipo *soft-input/soft-output*, isto é, recebem decisões suaves e passam decisões suaves também. A ideia central do processo de decodificação iterativa repousa no conceito de LLR extrínseca.

O processo de decodificação irá melhorar a decisão suave acerca de um *bit*, portanto após a decodificação, a LLR de um *bit* será alterada. Para um código sistemático, a LLR (decisão suave) na saída do decodificador é igual a

$$L(\hat{s}_t) = L'(\hat{s}_t) + L_e(\hat{s}_t),$$

em que $L'(\hat{s}_t)$ é a LLR de um *bit* após demodulação (entrada do decodificador) e $L_e(\hat{s}_t)$ é chamada de LLR extrínseca, e representa o conhecimento adicional que é obtido pelo processo de decodificação [9].

O primeiro decodificador, ao decodificar a sequência recebida, extrai informação adicional dos *bits* de paridade e corrige as decisões suaves recebidas pelo demodulador, modificando, dessa forma, as suas confiabilidades. A diferença entre as decisões suaves (LLRs) na saída e na entrada do decodificador dá-se o nome de LLR extrínseca. A LLR extrínseca fruto da primeira decodificação é utilizada para refinar a LLR *a priori* no processo de decodificação do segundo decodificador, que passará a possuir um ponto de partida mais confiável do que se a decodificação fosse feita sem modificação da LLR *a priori*. Assim como o primeiro decodificador, o segundo decodificador extrai informação de seus *bits* de paridade, que são estatisticamente independentes dos *bits* de paridade do primeiro decodificador devido ao entrelaçamento feito na codificação, e refina as decisões suaves mais uma vez. A LLR extrínseca fruto da segunda decodificação é calculada e usada para corrigir

a LLR *a priori* usada no primeiro codificador, que agora possui um ponto de partida ainda melhor. O processo iterativo continua até que se atinja um grau de confibialidade desejado ou um limite de iterações seja atingido. A Figura 3.10 ilustra o processo de decodificação iterativa.

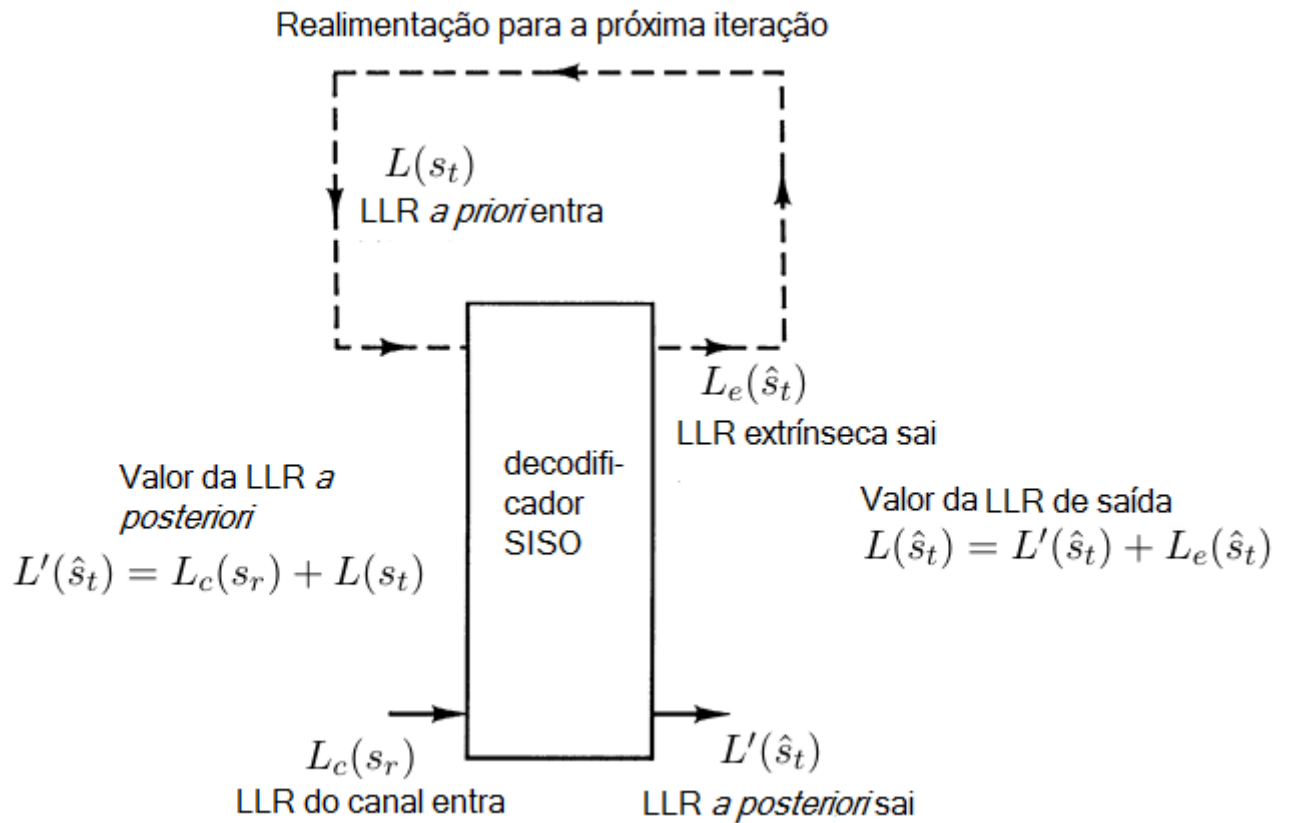


Figura 3.10: Decodificação Iterativa [9].

Capítulo 4

O bloco de codificação de canal e HARQ

Neste capítulo o bloco de codificação de canal, onde ocorre o processo de rate matching, do padrão LTE é abordado. O protocolo HARQ é explicado em detalhes, dentro do contexto do padrão LTE.

4.1 O bloco de codificação de canal do LTE

O bloco de codificação de canal recebe os *bits* de informação que devem ser enviados, aplica CRC à sequência de *bits*, segmenta a sequência em blocos caso seja necessário, aplica CRC em cada bloco, aplica codificação turbo em cada bloco, perfura os *bits* codificados a fim de obter a taxa de codificação desejada, concatena os blocos segmentados e envia os *bits* perfurados ao modulador. A Figura 4.1 mostra as etapas feitas no bloco de codificação.

4.1.1 CRC

Considere a sequência de *bits* de entrada do bloco de codificação $\{a_i\} = a_0, a_1, a_2, a_3, \dots, a_{A-1}$ em que A é o tamanho da sequência $\{a_i\}$. Sobre esta sequência de *bits* de entrada é calculada uma CRC de tamanho $L = 24$ cujo polinômio gerador é $g_{CRC24A}(D)$. Os vinte e quatro *bits* de paridade gerados pela CRC são acrescentados ao final da sequência $\{a_i\}$ e passam a ser chamados de $\{b_i\}$. O tamanho de $\{b_i\}$ é $B = A + L$.

4.1.2 Segmentação em Blocos

Conforme descrito na documentação oficial do 3GPP, se B , o tamanho de $\{b_i\}$, for superior a $Z = 6144$, a sequência deve ser segmentada em blocos menores [5]. Os blocos de *bits* podem ter até dois tamanhos diferentes, escolhidos dentre uma sequência crescente de tamanhos pré-determinados. Para que a diferença entre o tamanho dos blocos não seja muito expressiva, os dois

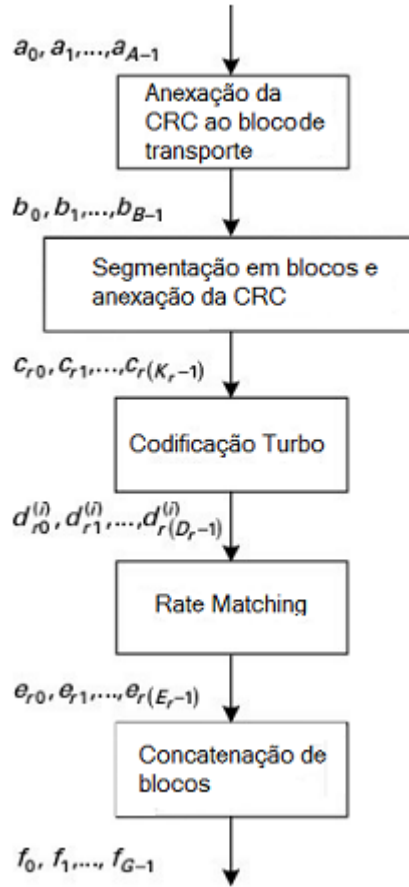


Figura 4.1: Bloco de codificação do LTE [3].

tamanhos escolhidos devem pertencer à sequência de tamanhos possíveis e serem consecutivos. Se for necessário, *bits* de preenchimento são adicionados no início do primeiro bloco.

A sequência de *bits* de saída após segmentação em blocos é chamada de $\{c_{ri}\} = c_{r0}, c_{r1}, c_{r2}, c_{r3}, \dots, c_{r(K_r-1)}$ em que r é o número do bloco e K_r é o número de *bits* que o bloco r possui.

Cada bloco é submetido ainda a um novo processo de codificação CRC. Os *bits* $c_{r0}, c_{r1}, c_{r2}, c_{r3}, \dots, c_{r(K_r-L-1)}$ são utilizados para gerar os L *bits* de paridade $p_{r0}, p_{r1}, p_{r2}, \dots, p_{r(L-1)}$ cujo polinômio gerador é $g_{CRC24B}(D)$. Esses *bits* de paridade são acrescentados ao fim da sequência de cada bloco. Para o cálculo da CRC, os *bits* de preenchimento são considerados iguais a 0, caso existam.

4.1.3 Codificação turbo

O codificador recebe os r blocos gerados no processo anterior e aplica codificação turbo em cada bloco. A sequência de entrada ao sub-bloco de codificação é denominada $\{c_i\} = c_0, c_1, c_2, c_3, \dots, c_{K-1}$. Existem r sequências $\{c_i\}$ - uma para cada bloco. Três sequências de *bits* são geradas pelo processo de codificação: a sequência de *bits* sistemáticos, a sequência de *bits* de paridade 1, relativa ao primeiro codificador, e a sequência de *bits* de paridade 2, relativa ao segundo codificador.

Os *bits* codificados são chamados de $\{d_i^j\} = d_0^j, d_1^j, d_2^j, d_3^j, \dots, d_{D-1}^j$ onde D é o número de *bits* de codificação por sequência de *bits* de saída e j indexa as sequências. $\{d_i^0\}$ é a sequência de *bits* sistemáticos, $\{d_i^1\}$ é a sequência de *bits* de paridade 1 e $\{d_i^2\}$ é a sequência de *bits* de paridade 2.

A terminação de treliça adiciona quatro *bits* a cada sequência, portanto o tamanho D de cada sequência é $D = K + 4$.

4.1.4 Rate Matching

No canal DL-SCH o padrão LTE utiliza códigos turbo cuja taxa é $1/3$ [5]. Como não há outro esquema de codificação definido para o canal DL-SCH, a princípio a taxa de codificação é sempre fixa: $1/3$.

O problema que isso gera é que a taxa de transmissão será sempre reduzida em um fator de $1/3$ em qualquer estado do canal de comunicação. Se, por exemplo, o canal está com baixa qualidade e tem-se uma razão sinal-ruído baixa a taxa de codificação de $1/3$ permite obter mais confiabilidade na comunicação e, conseqüentemente, menos erros que uma taxa alta. Neste caso a taxa $1/3$ está adequada ao estado do canal de comunicação e é muito útil para tal situação. Se, por outro lado, o canal está com boa qualidade a taxa de $1/3$ é desnecessária, pois com uma taxa de codificação maior poder-se-ia obter desempenho satisfatório. A taxa de codificação superior a $1/3$ (por exemplo $2/3$) é suficiente para garantir uma comunicação com uma taxa de *bits* errados aceitável e possui a importante vantagem de transmitir mais *bits* de informação por segundo.

Além de poder variar a taxa de codificação, o padrão LTE permite que o esquema de modulação seja modificado concomitantemente e automaticamente, escolhendo-se dessa forma o grau de robustez e a taxa de transmissão de *bits* adequada para cada situação. Esta técnica é chamada de *Adaptive Coding and Modulation* e tem o potencial de melhorar a eficiência espectral do sistema de comunicação.

Em um sistema que visa obter altas taxas de transmissão como o LTE, ter a taxa de codificação fixa é um problema. É preciso adaptar a taxa de codificação de acordo com o estado do canal para atingir-se as maiores taxas de transmissão de *bits* (de informação). Para resolver este problema os *bits* recém-codificados sofrem o processo chamado de perfuração (*puncturing*) que consiste em eliminar e, conseqüentemente, não transmitir certos *bits*. Isto é feito na etapa de *rate matching* que será descrita nesta seção.

Rate matching é a última etapa do processo de codificação de canal, em que os *bits* recém-codificados são perfurados e a taxa de codificação é definida. Graças ao *rate matching*, a taxa de codificação pode ser adaptada de acordo com a qualidade do canal de comunicação.

No padrão LTE, o *rate matching* consiste em 3 etapas: entrelaçamento, coleta de *bits* e seleção de *bits*. O *rate matching* trata as 3 sequências de *bits* provenientes do codificador turbo: a sequência de *bits* sistemáticos e as duas sequências de *bits* de paridade. Cada bloco de *bits* codificados é tratado individualmente e, ao final, os blocos pós - *rate matching* são concatenados e enviados ao modulador.

4.1.4.1 Entrelaçamento

No processo de entrelaçamento os *bits* de cada sequência de *bits* são embaralhados, isto é, os *bits* são permutados entre si de acordo com um vetor de permutação. Como exemplo ilustrativo seja a sequência de *bits* $x = [x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10}]$ e o vetor de permutação $P = [2 1 6 10 7 3 5 9 8 4]$. A sequência pós - permutação $x_p = [x_2 x_1 x_6 x_{10} x_7 x_3 x_5 x_9 x_8 x_4]$. É válido ressaltar que o processo é mais complicado que o exemplo dado e é descrito integralmente em [5].

O entrelaçamento é feito para espalhar erros contíguos e conferir mais robustez a erros em rajada melhorando o desempenho do decodificador [2].

4.1.4.2 Coleta de *bits*

Após entrelaçamento, tem-se 3 sequências de *bits* embaralhadas. A coleta de *bits* consiste em agrupar as 3 sequências em um *buffer* circular a fim de selecionar os *bits* na etapa seguinte. Um *buffer* circular é um *buffer* de dados em que considera-se que o último elemento (o último dado) e o primeiro elemento estão ligados como se fossem casas vizinhas. Desta forma o próximo elemento após o último elemento é o primeiro elemento e o "círculo" se fecha.

Primeiramente todos os *bits* da sequência sistemática são inseridos no começo do *buffer* circular. Em seguida são inseridos um a um os *bits* das sequências de paridade de forma entrelaçada, isto é, primeiro é inserido o primeiro *bit* da sequência de *bits* de paridade 1, depois é inserido o primeiro *bit* da sequência de *bits* de paridade 2, em seguida é inserido o segundo *bit* da sequência de *bits* de paridade 1 e assim por diante. A Figura 4.2 ilustra o processo.

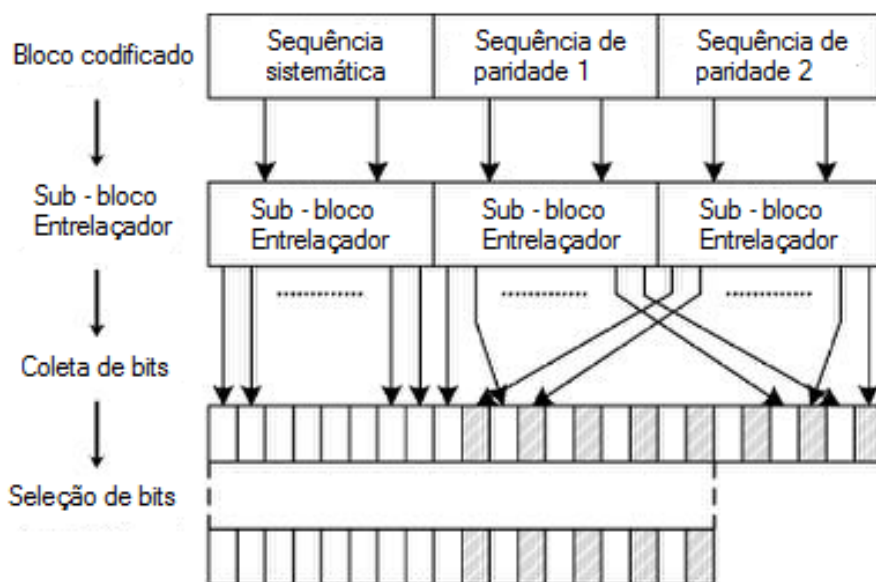


Figura 4.2: As 3 etapas de rate matching [3].

4.1.4.3 Seleção de *bits*

Tem-se agora um *buffer* circular contendo as 3 sequências de *bits* embaralhados. Nessa etapa, são selecionados os *bits* escolhidos para transmissão. Primeiramente é calculado, baseado no tamanho do bloco de transporte, no esquema de modulação e no modo MIMO o parâmetro E , que corresponde ao tamanho da sequência de *bits* de saída. Em seguida, é calculado o ponto de partida da leitura do *buffer* circular, k_0 , que depende da versão de redundância (*redundancy version*) da transmissão atual, que é diferente para diferentes retransmissões associadas ao processo de HARQ. São lidos do *buffer* circular, a partir do índice k_0 , E *bits*. Estes E *bits* são os *bits* selecionados para transmissão. Caso a leitura chegue ao final do *buffer*, ela é retomada do começo, devido a sua natureza circular, até que se tenha E *bits*.

Após seleção de *bits*, os *bits* provenientes de cada bloco são concatenados bloco a bloco, prontos para serem enviados ao modulador.

4.2 HARQ

HARQ (*Hybrid Automatic Repeat Request*) é a combinação de um código corretor de erros e o protocolo de repetição de pacotes de *bits* ARQ (*Automatic Repeat Request*) [3].

Um código corretor de erros é um esquema de codificação que por meio do acréscimo de *bits* de redundância à mensagem é capaz de corrigir erros nas palavras - código recebidas pelo receptor. No caso do canal DL-SCH do padrão LTE é utilizada a codificação turbo.

O protocolo ARQ consiste em, utilizando um código detector de erros como por exemplo o *Cyclic Redundancy Check* (CRC), verificar se os pacotes de dados decodificados estão livres de erros e caso existam erros, solicitar ao transmissor que retransmita o pacote corrompido. O protocolo ARQ estabelece a maneira com a qual o pedido de retransmissão é realizado. No protocolo ARQ, o pacote corrompido é descartado e espera-se que o pacote retransmitido venha sem erros. Caso contrário, uma nova retransmissão é solicitada. O processo ocorre até que o pacote seja recebido sem erros ou que um limite no número de retransmissões seja atingido.

Os protocolos ARQ podem ser classificados em 3 categorias: *stop-and-wait* (SAW), *Go-Back-N* e *selective repeat protocols* [3].

4.2.1 Stop-and-wait

Stop-and-wait é o tipo de protocolo ARQ mais simples entre os 3 citados. Neste esquema, o transmissor envia um pacote e espera o retorno do receptor. O receptor, ao receber o pacote, verifica-o para ver se há erros. Se não há erros, o receptor envia um *acknowledgment* positivo (ACK) ao transmissor. Um *acknowledgment* é uma simples mensagem que indica se a mensagem foi recebida ou não. Caso haja erros no pacote, o receptor envia um *acknowledgment* negativo (NACK). Ao receber o retorno do receptor, caso tenha recebido um NACK, o transmissor reenvia o pacote recém - transmitido ao receptor e espera um novo retorno. Caso contrário, o transmissor

envia um novo pacote. Se o retorno do receptor não for recebido após um certo tempo de espera, o transmissor supõe que o pacote não foi recebido e retransmite o pacote.

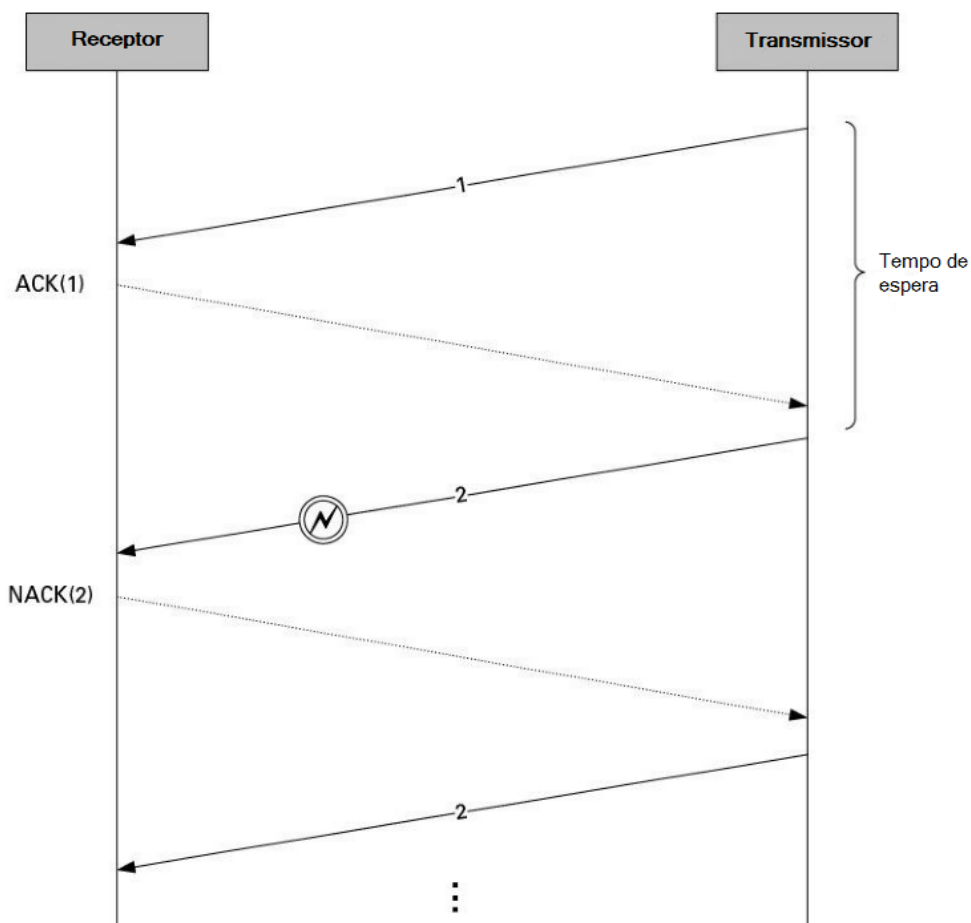


Figura 4.3: ARQ *Stop-and-wait* [3].

Um problema do protocolo *stop-and-wait* é o grande intervalo de tempo entre transmissões (atraso). Isso ocorre porque enquanto o transmissor não receber o *acknowledgment* do receptor ele fica ocioso e o tempo de espera pode ser demasiadamente longo. No sistema LTE, o tempo de transmissão de um pacote dura um *subframe* (1 ms), mas o tempo de espera para retransmissão de pacotes são 7 *subframes*, isto é, o transmissor ficaria 7 ms ocioso até receber o *acknowledgment* do receptor [3].

4.2.2 Go-Back-N

O *Go-Back-N* é um protocolo ARQ que tenta resolver o problema do tempo de espera do protocolo *Stop-and-wait*. Neste protocolo, o transmissor envia N pacotes ao receptor sem esperar as *acknowledgments*, em vez de enviar apenas um pacote. Quando um pacote é recebido e contém erros, o receptor envia um NACK ao transmissor e ignora os próximos pacotes da sequência. Isso garante que os pacotes sejam recebidos em ordem. Ao receber o NACK, o transmissor reinicia a

transmissão da sequência de pacotes a partir do pacote que contém erros. A Figura 4.4 ilustra o processo:

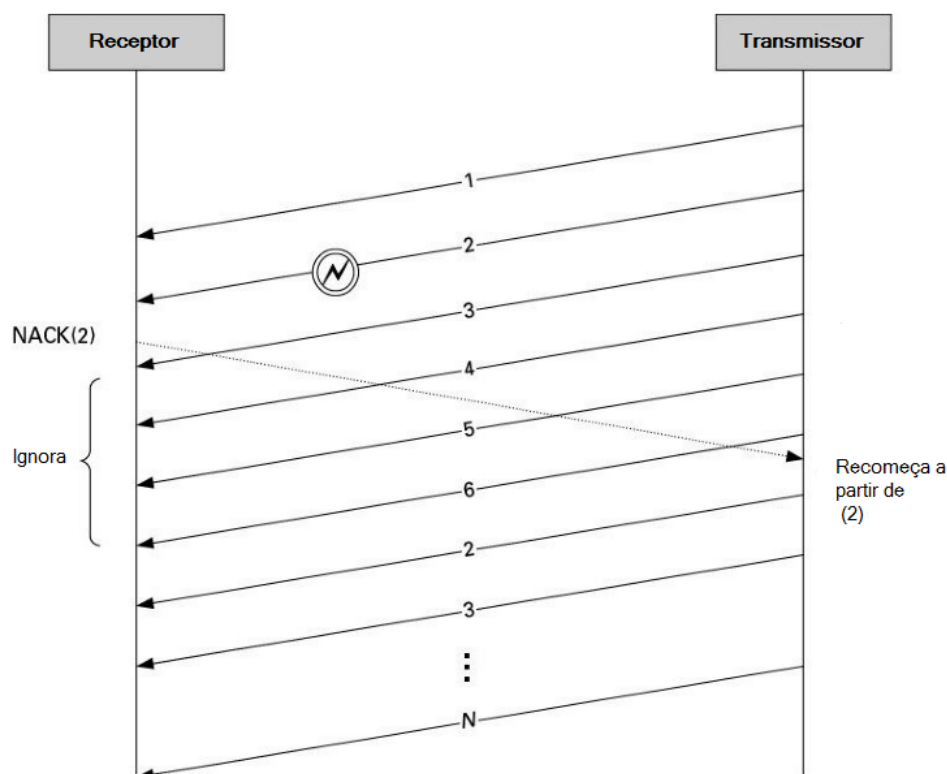


Figura 4.4: ARQ *Go-back-N* [3].

O *Go-Back-N* resolve o problema do tempo de espera entre transmissões, mas no processo surge um outro problema: como o receptor ignora os pacotes recebidos após o envio de um NACK, o transmissor é forçado a retransmitir pacotes que foram ignorados e que poderiam estar sem erros.

4.2.3 Selective Repeat

No protocolo *ARQ selective repeat*, o transmissor envia N pacotes continuamente sem esperar o retorno do receptor, assim como no protocolo *Go-Back-N*. Os N pacotes configuram uma "janela", isto é, um conjunto de N pacotes que são esperados a ser transmitidos e recebidos. Quando um pacote é recebido com sucesso, o receptor desloca a sua janela em 1 pacote. Por exemplo, ao iniciar a transmissão a janela de N pacotes se encontra entre o primeiro e o n -ésimo pacote. Após o correto recebimento do primeiro pacote, a janela do receptor é deslocada em um pacote e passa a se encontrar entre o segundo pacote e o n -ésimo primeiro pacote indicando que os pacotes que o receptor espera receber vão do segundo ao n -ésimo primeiro pacote. Quando o transmissor recebe um ACK, ele, por sua vez, desloca em um pacote a sua janela. Por exemplo, ao início da transmissão, a janela do transmissor está entre o primeiro e o n -ésimo pacote. Após recebimento de ACK para o primeiro pacote, o transmissor desloca a sua janela, que passa a conter do segundo ao n -ésimo primeiro pacote. O processo vai, dessa forma, se desenvolvendo até que todos os

pacotes tenham sido transmitidos.

Quando há um erro em um pacote recebido, o receptor envia um NACK ao transmissor. Diferentemente do protocolo *Go-Back-N*, o receptor continua a aceitar pacotes após o recebimento do pacote corrompido e o transmissor apenas retransmite os pacotes que não foram bem - recebidos pelo receptor (os que contem erros). Para realizar isto, é necessário que em seu *feedback* o receptor indique ao transmissor quais pacotes ele deve retransmitir. Isso pode ser feito por meio de um vetor de *bits* de tamanho N em que cada *bit* representa o ACK/NACK de um pacote. Cada *feedback* do receptor contém esse vetor, que é atualizado a cada retransmissão. Com isso, a mensagem de *feedback* pode ficar bem grande, se o tamanho da janela também o for, e isto pode ser um problema [3].

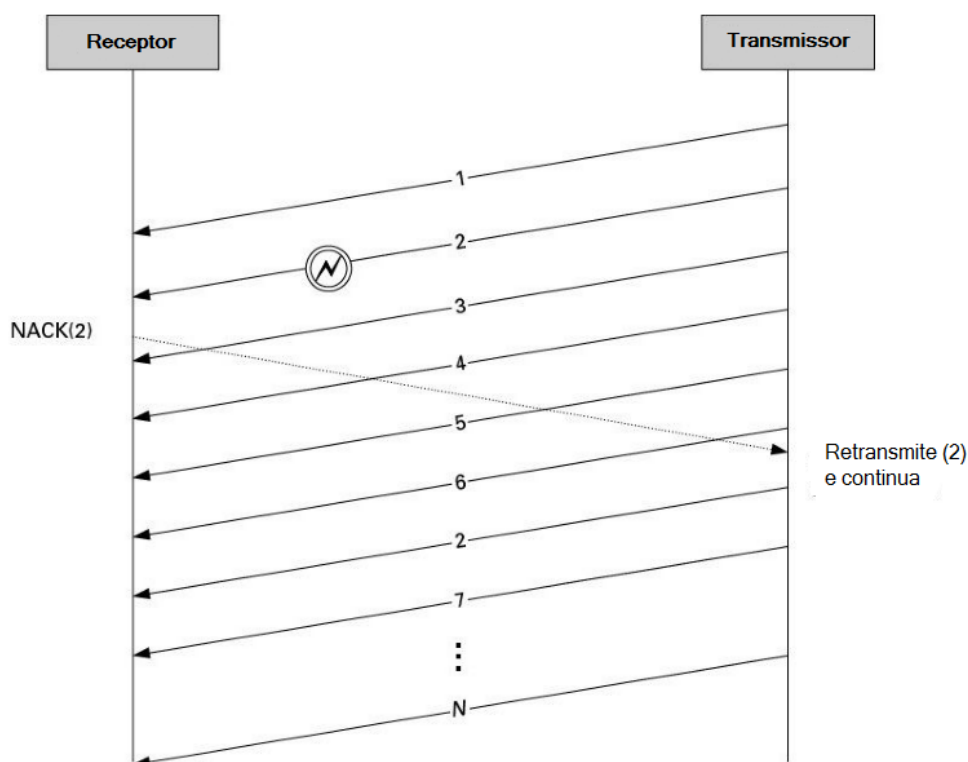


Figura 4.5: ARQ *Selective Repeat* [3].

O receptor recebe e armazena os pacotes bem-recebidos enquanto uma nova retransmissão dos pacotes corrompidos não chega. Para não enviar às camadas superiores pacotes fora de ordem, o receptor deve armazenar os pacotes, o que implica que ele deve possuir um grande *buffer* de armazenamento para lidar com esse problema. Por exemplo, na figura 4.6, o pacote 3 não poderia ser repassado à uma camada superior até que o pacote 2 seja bem-recebido e, conseqüentemente, teria que ser armazenado. Caso o *buffer* de recepção do receptor não seja suficientemente grande, os pacotes recebidos podem ser perdidos [3].

Para concluir esta seção, destaca-se que o ARQ *selective repeat* resolve os problemas dos dois protocolos anteriores, porém exige que o receptor tenha um grande *buffer* de recepção para armazenar os pacotes recebidos fora de ordem e impõe que a mensagem de *acknowledgement*

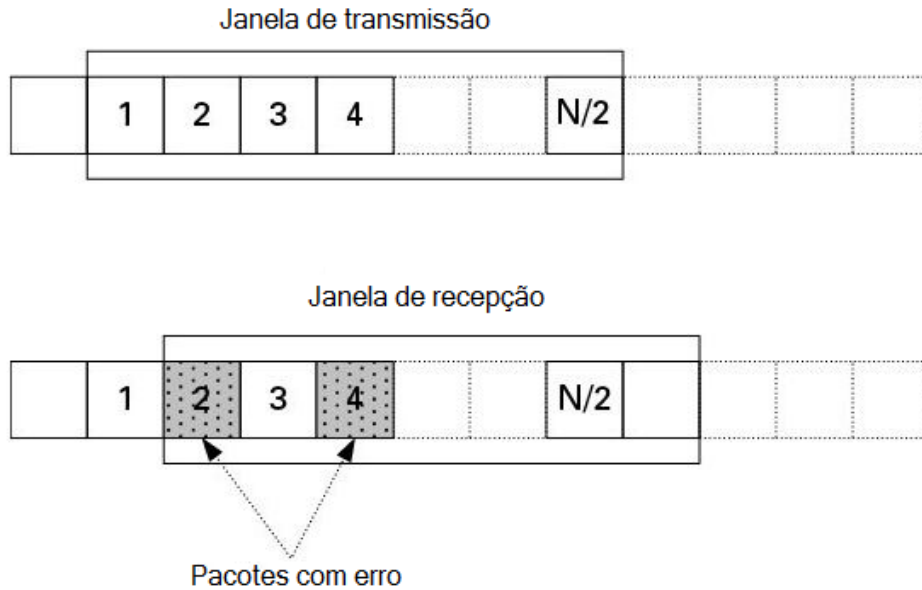


Figura 4.6: Janelas de recepção e transmissão do ARQ *Selective Repeat* [3].

contenha também o número do pacote a ser retransmitido, em contraste com o protocolos mais simples *stop-and-wait*.

4.2.4 N-channel stop-and-wait

O protocolo *N-channel-stop-and-wait* é uma variação do protocolo *stop-and-wait*. A idéia nova deste protocolo é agendar a transmissão de *acknowledgment* entre transmissor e receptor para eliminar as desvantagens do protocolo *Selective Repeat*, conservando seus benefícios. O processo ocorre da seguinte maneira: O transmissor envia continuamente, sem esperar *acknowledgment*, N pacotes ao receptor. O receptor, ao receber um pacote, verifica se há erros. O receptor então envia um simples ACK/NACK ao transmissor, sem informação de qual pacote pertence o *feedback*, em um tempo pré-determinado e fixo. O transmissor sabe quando a mensagem de *acknowledgment* relativo a cada pacote vai chegar, pois os *acknowledgments* foram previamente agendados e, dessa forma, não há necessidade de envio do número do pacote na mensagem de *acknowledgment*.

O número N de pacotes a serem enviados é escolhido de maneira inteligente, de modo a garantir que o transmissor não fique ocioso. O efeito final é a impressão de que há N canais em paralelo empregando o protocolo *stop-and-wait*. A Figura 4.7 ilustra o processo.

O protocolo *N-channel-stop-and-wait* possui os benefícios do protocolo *selective repeat* e soluciona as desvantagens, isto é, não requer um grande *buffer* de recepção e não é preciso enviar uma longa mensagem de *acknowledgment*. Devido a sua simplicidade e desempenho, é o protocolo usado no sistema LTE. [3]

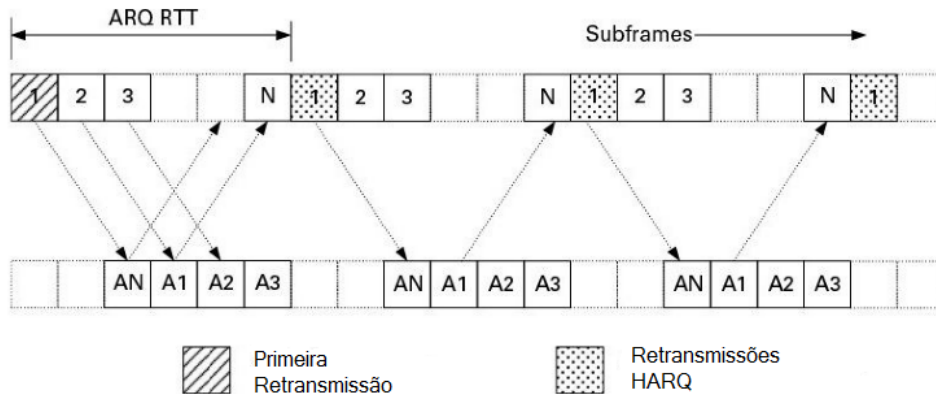


Figura 4.7: ARQ N -channel stop-and-wait [3].

4.2.5 Tipos de HARQ

A ideia central do HARQ é poder reaproveitar, de alguma forma, a informação contida nos pacotes recebidos corrompidos, em vez de simplesmente descartá-los. Existem 3 tipos de HARQ: HARQ tipo 1, HARQ tipo 2 e HARQ tipo 3 [10].

4.2.5.1 HARQ tipo 1

A forma mais simples de HARQ consiste em utilizar um código corretor de erros para corrigir os erros recebidos no receptor. Caso isto falhe, o código detector de erros acusará que há erros no pacote e que este está corrompido. O pacote recebido, entretanto, não é descartado e um novo pacote contendo os mesmos *bits* já enviados é solicitado utilizando algum protocolo ARQ. Esta versão de HARQ é quase igual ao ARQ puro. A diferença é que, no HARQ tipo 1, um código corretor de erros é utilizado e o pacote corrompido não é descartado.

A razão para o pacote corrompido não ser descartado é que, utilizando alguma técnica de combinação de pacotes conjuntamente com o HARQ tipo 1, pode-se extrair mais informação dos pacotes, o que aumenta a probabilidade de decodificar corretamente o pacote recebido.

A ideia de combinar os pacotes retransmitidos foi, primeiramente, proposta por Chase em 1985 [11] e por isso o HARQ tipo 1 é, às vezes, chamado de *Chase Combining*. Para tornar o texto claro, quando o HARQ tipo 1 é usado em conjunto com combinação de pacotes, ele será referido como HARQ tipo 1 com *Chase Combining*. O HARQ tipo 1 com *Chase Combining* pode ser visto como uma mistura entre o código-corretor de erros usado e um simples código de repetição. Cada retransmissão pode ser vista como se a taxa do código de repetição fosse incrementada. Há, com isso, um ganho de diversidade, pois como cada pacote é transmitido em um estado diferente de canal, tem-se diferentes SNRs (*signal-to-noise ratio*) para cada pacote.

A combinação de pacotes é feita usando as decisões suaves (LLRs dos *bits*). A cada retransmissão, recebe-se uma nova decisão suave a respeito de cada *bit* retransmitido. Em seguida, é preciso combinar de alguma forma essas decisões suaves referentes ao mesmo *bit*. Existem diversas manei-

ras de combinar as decisões suaves dos pacotes recebidos. Por exemplo, poder-se-ia fazer a média entre as LLRs recebidas, somar as LLRs recebidas ou escolher a maior LLR entre as recebidas. Neste trabalho, optou-se por somar as LLRs recebidas como técnica de combinação de pacotes ou *Chase Combining*. Esta escolha é baseada na referência [12] em que é apontado que somar as LLRs recebidas resulta em melhor desempenho em relação às outras técnicas citadas.

O HARQ tipo 1 não exige que o receptor tenha um grande *buffer* para armazenar as retransmissões em comparação a outros tipos de HARQ.

4.2.5.2 HARQ tipo 2

O HARQ tipo 2, também chamado de redundância incremental, é uma forma mais sofisticada de retransmissão de pacotes. A diferença em relação ao HARQ tipo 1 é que os *bits* retransmitidos não são os mesmos *bits* que foram previamente enviados. Ao receber um pedido de retransmissão, o transmissor envia *bits* de paridade que não estavam presentes na primeira transmissão. Com o conhecimento de mais *bits* de paridade, o receptor tenta decodificar a sequência de *bits* recebida com mais informação que anteriormente, pois há mais *bits* de paridade. Isto é equivalente a diminuir a taxa de codificação, o que implica aumentar a probabilidade de decodificação correta, pois agora o código é mais forte.

Como exemplo, pode-se pensar em um esquema HARQ tipo 2 cujo código corretor de erros é igual ao do canal DL-SCH do padrão LTE: código turbo com taxa de codificação é $1/3$. Supondo que se quer transmitir k *bits* de informação, o total de *bits* após codificação turbo será igual a $3k$, k *bits* sistemáticos e $2k$ *bits* de paridade. O protocolo poderia funcionar da seguinte maneira: na primeira transmissão, são enviados os k *bits* sistemáticos e $1/4$ dos *bits* de paridade, isto é, $k/2$ *bits* de paridade. Caso não seja possível decodificar corretamente o pacote na segunda retransmissão, são enviados mais $k/2$ *bits* de paridade. Caso haja nova falha de decodificação, mais $k/2$ *bits* de paridade são enviados, até que a decodificação seja bem-sucedida ou até que o número de retransmissões ultrapasse o número máximo de retransmissões.

Nota-se que, no HARQ tipo 2, as retransmissões podem não ser decodificadas sem que se utilize outras retransmissões. No exemplo dado, apenas a primeira transmissão poderia ser decodificada sem necessidade das outras, contrastando com os outros dois tipos de HARQ, 1 e 3, em que todas as retransmissões podem ser decodificadas sem depender de pacotes transmitidos anteriormente.

O HARQ tipo 2 tem melhor desempenho que o HARQ tipo 1 [3]. Em contrapartida, o HARQ tipo 2 exige que o receptor possua um grande *buffer* de recepção, o que pode inviabilizar seu emprego.

4.2.5.3 HARQ tipo 3

Assim como no HARQ tipo 2, os pacotes retransmitidos contêm *bits* diferentes dos *bits* dos pacotes enviados anteriormente, mas neste tipo de HARQ os pacotes retransmitidos são auto-decodificáveis, ou seja, em cada pacote é enviada quantidade de informação suficiente de modo

que todos os *bits* transmitidos podem ser decodificados sem precisar do conteúdo de outro pacote já transmitido.

O HARQ tipo 3 está intimamente ligado ao processo de *rate matching*. A seleção dos *bits* escolhidos para serem transmitidos é função do HARQ tipo 3. Por exemplo, supõe-se que após codificação turbo, tenha-se $3k$ *bits*, k *bits* sistemáticos e $2k$ *bits* de paridade. No LTE, existem 4 versões de redundância, isto é, os $3k$ *bits* são usados para formar 4 conjuntos de *bits* diferentes, cada conjunto contendo E *bits*. É válido destacar que é possível que um mesmo *bit* esteja presente em mais de um conjunto. Na primeira transmissão, envia-se o primeiro conjunto de *bits*. Caso haja falha na decodificação, envia-se o segundo conjunto de *bits*. Caso haja nova falha na decodificação, envia-se o terceiro conjunto de *bits* e assim por diante, diminuindo-se a taxa de codificação do código assim como no HARQ tipo 2.

Como os 4 conjuntos de *bits* gerados no *rate matching* podem conter alguns *bits* iguais, é possível que em uma dada retransmissão, o receptor receba *bits* repetidos. Quando isso ocorre, é possível combinar as decisões suaves a respeito desses *bits* por meio de *Chase Combining*, assim como feito no HARQ tipo 1.

Para empregar o HARQ tipo 3, o receptor necessita de um grande *buffer* de recepção para armazenar os pacotes retransmitidos, configurando uma desvantagem em relação ao HARQ tipo 1 e, possivelmente, 2.

É o esquema de HARQ utilizado no canal DL-SCH do padrão LTE [3].

Capítulo 5

Simulação

Neste capítulo, é explicado como o simulador funciona e são apresentados os resultados da simulação.

5.1 O simulador WiSiL

O WiSiL (*wireless simulation lab*) é um simulador de enlace direto da camada física do sistema LTE. Ele foi desenvolvido majoritariamente em linguagem MATLAB pela Universidade de Brasília, pelo Instituto Nokia de Tecnologia e pela Universidade Federal do Rio de Janeiro. Alguns de seus recursos são: simulação correta dos quadros do LTE, com sinais de referência, modo MIMO, codificação turbo e utilização de modelos de canal adequados para simulação do canal sem fio.

A Figura 5.1 mostra o diagrama de blocos da cadeia de comunicação feita pelo simulador WiSiL.

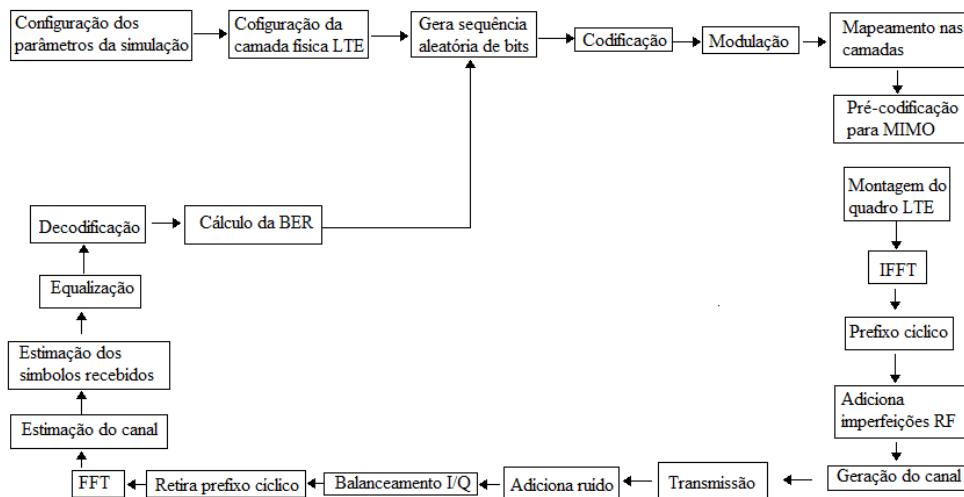


Figura 5.1: Diagrama de Blocos do simulador WiSiL [13].

O foco deste trabalho está nos blocos de codificação e decodificação. O bloco de codificação recebe como entrada a sequência de *bits* de informação que se deseja transmitir. Primeiramente aplica-se CRC à sequência de *bits* de entrada. Se for necessário, a sequência de *bits* de entrada

é dividida em sequências menores, chamadas de blocos. Aplica-se nova CRC a cada bloco e, em seguida, aplica-se codificação turbo em cada bloco. Por fim, cada bloco passa pelo processo de *rate matching*. A saída do codificador é enviada ao modulador digital.

O bloco de decodificação recebe como entrada as decisões suaves, passadas pelo estimador de símbolos, a respeito de cada *bit* de cada bloco recebido. Na decodificação, é desfeito o *rate matching*, o que organiza as decisões suaves dos *bits* recebidos para serem decodificadas pelo decodificador turbo. A decisão suave de um *bit* que não foi transmitido é considerada igual a zero no decodificador, pois LLR igual a zero significa máxima incerteza na decisão. Se for uma retransmissão, as decisões suaves recebidas são combinadas com as decisões suaves recebidas previamente, caso contrário, após desfeito o *rate matching* as decisões suaves são enviadas diretamente ao decodificador turbo. Os *bits* são decodificados e em seguida é aplicado o teste CRC em cada bloco recebido. Se há falha no teste, é solicitada retransmissão do bloco recebido, de acordo com o protocolo de HARQ escolhido.

A simulação é organizada em *radio frames*, isto é, simula-se quadros inteiros do LTE. Ao final de cada simulação, calcula-se a taxa de *bits* errados. É possível também simular várias vezes o mesmo quadro, para minimizar variações aleatórias causadas pelo modelo do canal. Primeiramente define-se quantos quadros serão simulados. Em seguida, define-se quantas vezes a simulação de cada quadro será repetida.

O simulador possui uma série de parâmetros que são determinados pelo usuário para controlar a simulação. Para ter-se maior clareza na exposição dos resultados, alguns parâmetros serão discutidos nessa seção.

Largura de banda, prefixo cíclico e espaçamento entre subportadoras: São parâmetros relacionados ao OFDM e determinam a quantidade de elementos de recurso que será utilizada. Com efeito, estes parâmetros determinam a quantidade de símbolos que será transmitida em um quadro do LTE.

Tipo de canal: O simulador WiSiL permite o uso do modelo de canal 3GPP SCM *3GPP Spatial Channel Model*, cujo código em linguagem MATLAB utilizado nessa simulação foi desenvolvido pelo consórcio Winner. Também pode-se optar pelo uso de um simples canal AWGN.

Modulação: Determina o esquema de modulação utilizado e portanto a quantidade de *bits* por símbolo. Três esquemas de modulação são permitidos: QPSK, 16-QAM e 64-QAM.

Taxa de codificação: É a taxa de codificação do codificador de canal. Taxa de codificação igual a $1/3$ significa que todos os *bits* gerados pelo codificador turbo são transmitidos. Taxa de codificação igual a 1 significa que apenas $1/3$ dos *bits* gerados pelo codificador turbo serão enviados. A seleção de *bits* é feita no processo de *rate matching*.

Habilitação de HARQ: parâmetro que determina o uso ou não de HARQ.

Tipo de HARQ: determina qual tipo de HARQ que será utilizado: HARQ tipo 1 ou HARQ tipo 3.

Número de laços: Determina a quantidade de vezes que a simulação de um mesmo quadro

será repetida.

Número de quadros: Determina a quantidade de quadros (*radio frames*) que será simulada.

Número de antenas transmissoras, número de antenas receptoras, modo de transmissão MIMO, modo de recepção MIMO: Determinam como é feito o MIMO, quantas antenas são usadas, e se é utilizado *beamforming* ou diversidade de transmissão, por exemplo.

Algoritmo estimador de canal: É o algoritmo usado para estimar os coeficientes do canal. Há três técnicas de estimação de canal disponíveis: LS (*least squares*), MMSE (*minimum mean square error*) e MMSE robusto. Também é possível assumir que a estimação é ideal, ou seja, que o receptor conhece exatamente o canal.

Codificador: É o código corretor de erros usado na codificação de canal. Pode-se optar por utilizar o codificador turbo ou não utilizar codificação de canal.

Número de iterações do decodificador: É o número de iterações feito pelo decodificador turbo para refinar as decisões suaves.

5.2 Resultados

Os resultados apresentados neste trabalho são frutos do desenvolvimento de funções que complementam o simulador WiSiL para que ele produza uma simulação mais aderente ao padrão. O processo de *Rate Matching*, descrito na seção 2.7.4, foi implementado, adequando o simulador à especificação do 3GPP TS 36.212 Versão 12.10. Um processo de *de-Rate*, isto é, de desfazer o *Rate Matching* no decodificador também foi desenvolvido, testado e validado. Com isso, foi possível acrescentar a opção de variar a taxa de codificação da simulação, variando o número de *bits* perfurados.

O código detector de erros do sistema LTE, a CRC, também foi implementado no bloco de codificação. Isto possibilitou a implementação do HARQ tipo 1 com *Chase Combining* e HARQ tipo 3, dois protocolos de repetição de pacotes usados no sistema LTE.

É importante ressaltar que o simulador WiSiL realiza a simulação por *radio frame* do LTE. Entretanto, a retransmissão de pacotes HARQ é feita a cada *sub-frame* recebido e não após recepção de um *radio frame*, isto é, no contexto do HARQ um pacote corresponde ao conteúdo transmitido dentro de um *sub-frame* do LTE. Para contornar essa particularidade, decidiu-se aproveitar a segmentação em blocos feita no bloco de codificação da sequência de *bits* de mensagem. Considera-se que cada bloco equivale a um *sub-frame* e o HARQ é implementado então por blocos.

5.2.1 Variação da taxa de codificação

Para as simulações de *Rate Matching* os parâmetros mostrados na Tabela 5.2.1 são mantidos constantes.

As Figuras 5.2 e 5.3 mostram quatro curvas da taxa de *bits* errados (*Bit Error Rate* (BER)) em

Parâmetros	Valor
Largura de Banda	10 MHz
Espaçamento entre subportadoras	15KHz
Prefixo Cíclico	curto
Número de antenas transmissoras	1
Modo de transmissão MIMO	Nenhum
Algoritmo estimador de canal	LS
Número de antenas receptoras	1
Modo de recepção MIMO	Nenhum
Modulação	QPSK
Codificador	Turbo
Número de iterações do decodificador	5
HARQ	desabilitado
Imperfeições de RF	Não há
Quadros simulados	10
Laços simulados	5

Tabela 5.1: Parâmetros mantidos constantes durante a simulação de *rate matching*.

função da SNR(*signal-to-noise ratio*) por *bit* (E_b/N_0) para quatro diferentes taxas de codificação. Há duas diferenças nos parâmetros da simulação entre as duas figuras. A primeira é que as curvas da Figura 5.2 foram geradas utilizando o canal AWGN e as curvas da Figura 5.3 foram geradas utilizando o modelo de canal 3GPP SCM. O modelo 3GPP SCM é explicado detalhadamente em [14]. A segunda é que na Figura 5.2, a E_b/N_0 varia de 0 dB a 7 dB, e na Figura 5.3, a E_b/N_0 varia de 0 dB a 11 dB. Em ambos os gráficos, vê-se que em todas as curvas, a BER cai quando se aumenta a E_b/N_0 , mas a queda é mais acentuada quando é utilizada uma taxa de codificação menor. Isto ocorre porque quanto mais redundância é inserida, isto é, quanto menor a taxa de codificação, maior é a capacidade corretora do código e, naturalmente, mais erros são corrigidos quando as taxas de codificação são pequenas.

Na Figura 5.2, para a taxa de codificação $R = 0,75$, a BER chega a 0 a partir de $E_b/N_0 = 5$ dB para o número de *bits* simulado, e os pontos seguintes são omitidos. O mesmo ocorre para $R = 0,5$, a partir de $E_b/N_0 = 3$ dB e para $R = 1/3$, a partir de $E_b/N_0 = 2$ dB. Nesta simulação, é possível perceber o bom desempenho dos códigos turbo no canal AWGN, sendo capazes de levar a BER para valores inferiores a 10^{-6} para certos valores da SNR.

Na Figura 5.3 percebe-se o efeito nocivo do canal sem fio, modelado nesta simulação pelo canal 3GPP SCM. O desempenho dos códigos turbo no canal 3GPP SCM é consideravelmente inferior ao desempenho no canal AWGN. Isto ocorre porque o canal 3GPP SCM modela um canal sem fio real, e leva em consideração problemas de desvanecimento de sinal desconsiderados pelo canal AWGN.

Em ambas as figuras, nota-se também que, para SNRs baixas, não há ganho de codificação. Conclui-se da análise das duas figuras que o *rate matching* possibilita a variação das taxas de codi-

ficação que, por sua vez, permitem o emprego de técnicas de modulação e codificação adaptativas.

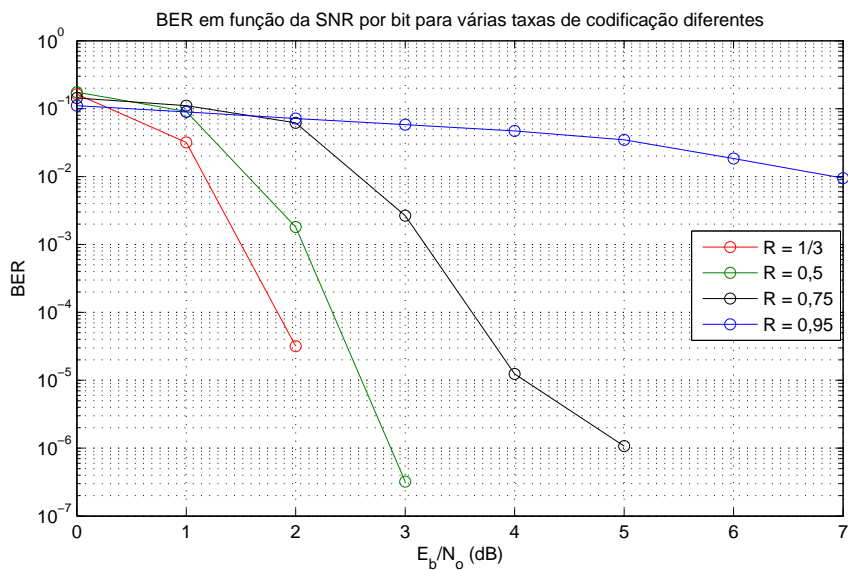


Figura 5.2: Ber em função da SNR por *bit* para várias taxas de codificação - canal AWGN.

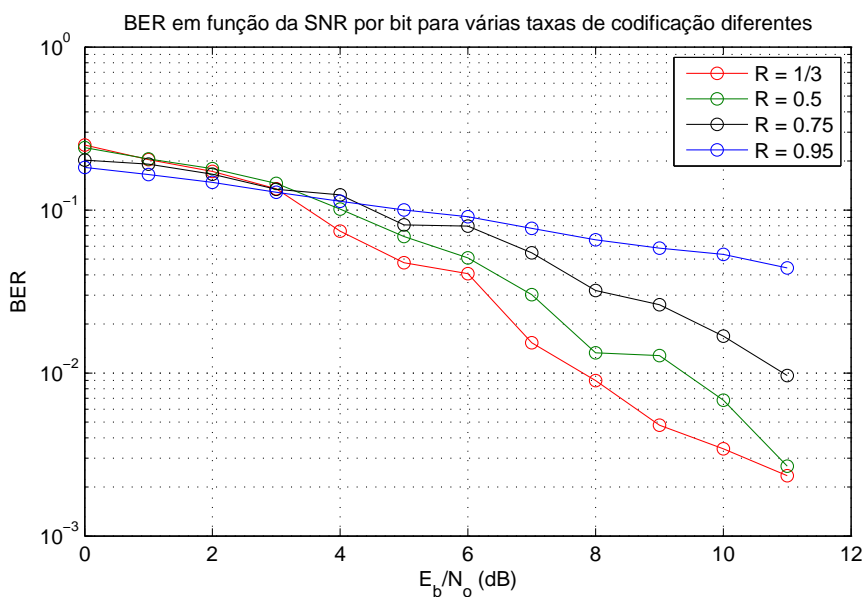


Figura 5.3: Ber em função da SNR por *bit* para várias taxas de codificação - canal 3GPP SCM.

5.2.2 HARQ

O HARQ é implementado da seguinte maneira. Primeiramente, um *radio frame* do LTE é montado e enviado. É válido lembrar que os *bits* de informação foram divididos em blocos no processo de segmentação de blocos e que, para esta simulação, considera-se que o HARQ é aplicado sobre os blocos. No decodificador, os *bits* de CRC indicam se houve erro em algum bloco. Se houve erro, o bloco é retransmitido (na prática o *radio frame* inteiro é retransmitido, mas a

combinação de pacotes é feita apenas para os blocos onde houve erros), combinado com o bloco recebido anteriormente, e o bloco resultante de sua combinação é decodificado. Caso ainda existam erros (constatados pela CRC), uma nova retransmissão é feita até que o bloco seja corretamente decodificado ou até que um limite máximo de retransmissão seja atingido.

Os parâmetros da Tabela 5.2.2 foram mantidos constantes para as simulações HARQ.

Parâmetros	Valor
Largura de Banda	10 MHz
Espaçamento entre subportadoras	15KHz
Prefixo Cíclico	curto
Canal	3GPP SCM
Número de antenas transmissoras	1
Modo de transmissão MIMO	Nenhum
Algoritmo estimador de canal	LS
Número de antenas receptoras	1
Modo de recepção MIMO	Nenhum
Modulação	QPSK
Codificador	Turbo
Número de iterações do decodificador	5
HARQ	habilitado
Imperfeições de RF	Não há
Taxa de codificação	0.75
Quadros simulados	20
Laços simulados	5

Tabela 5.2: Parâmetros mantidos constantes durante a simulação HARQ.

5.2.2.1 HARQ tipo 1 com Chase Combining

A implementação do HARQ tipo 1 com *Chase Combining* (ou combinação de pacotes) consiste em simplesmente somar, após demodulação e antes de entregar as decisões suaves ao bloco de decodificação, a LLR recebida com a LLR do pacote anterior. Isto funciona, pois somar as LLRs dos pacotes recebidos atua como um refinamento das decisões suaves, e tende a fornecer melhores condições para o decodificador [12].

A Figura 5.4 mostra o benefício à taxa de *bits* errados que o HARQ tipo 1 com *Chase Combining* pode trazer.

Ao comparar as duas curvas na Figura 5.4, vê-se que a combinação de pacotes melhora substancialmente a BER. É interessante notar que o ganho fica mais expressivo com o aumento da E_b/N_0 .

A seguir, na Figura 5.5, avalia-se como a BER muda com o aumento do limite máximo de retransmissão.

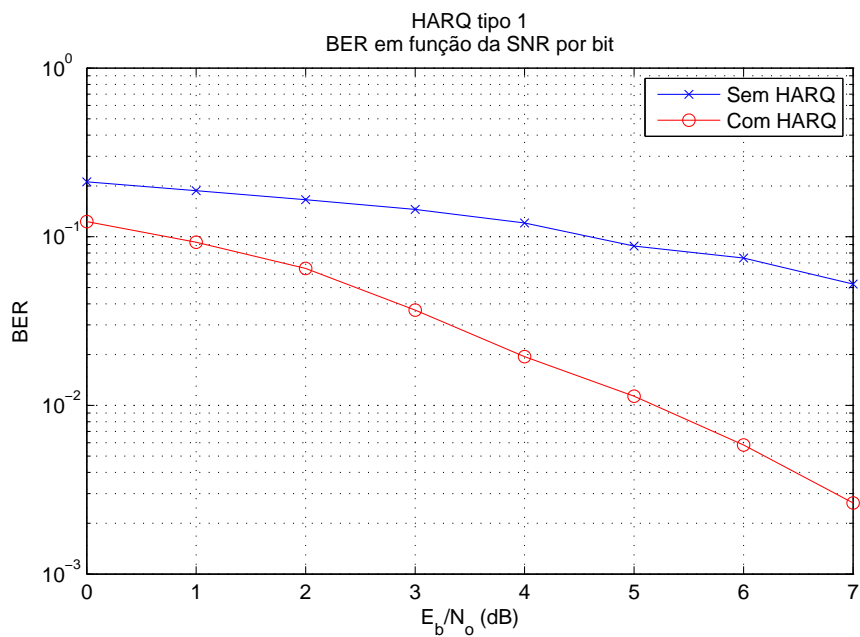


Figura 5.4: BER em função da SNR por *bit* para 1 retransmissão (ou seja sem HARQ) e 2 retransmissões HARQ tipo 1.

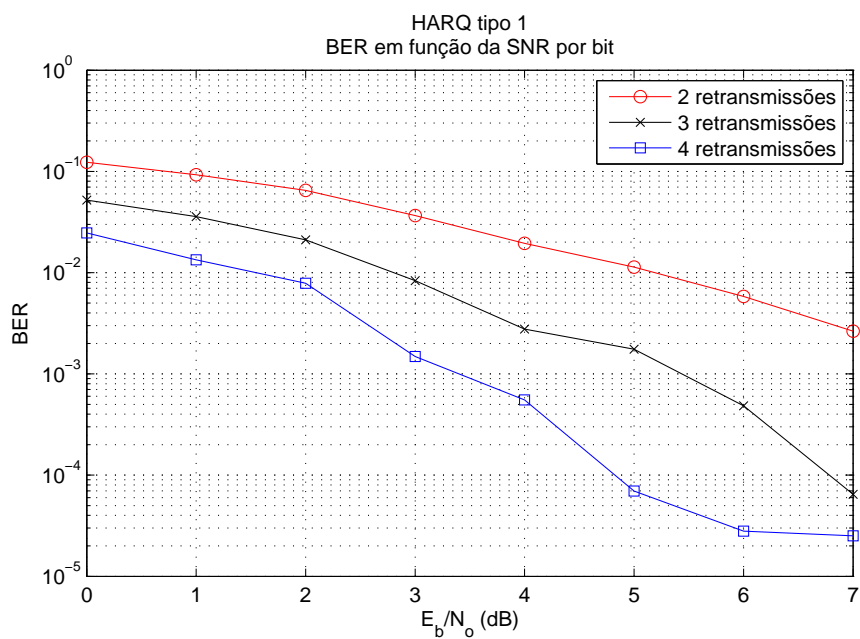


Figura 5.5: HARQ tipo 1 - BER em função da SNR por *bit* para sucessivas retransmissões.

A cada retransmissão, mais confiabilidade é acrescentada às decisões suaves resultantes da combinação de pacotes. Como era de se esperar, a BER melhora a cada retransmissão e na quarta retransmissão, a partir de $E_b/N_0 = 5$ dB, atingi-se erro inferior a 10^{-4} .

5.2.2.2 HARQ tipo 3

O processo de *rate matching* torna a implementação de HARQ tipo 3 bastante natural. Assim como no HARQ tipo 1, caso a CRC constata erros no bloco, ele é retransmitido, mas desta vez com uma versão de redundância diferente. Cada versão de redundância possui *bits* de paridade diferentes (apesar de haver alguma superposição, especialmente para taxas de codificação baixas). Há 4 versões de redundância, o que significa que, a partir da quarta retransmissão, passa a haver uma repetição total dos *bits*, isto é, nenhum *bit* contendo nova informação é enviado e o HARQ tipo 3 se transforma em uma mera combinação de pacotes.

A Figura 5.6 mostra a melhoria à taxa de *bits* errados que o HARQ tipo 3 pode trazer.

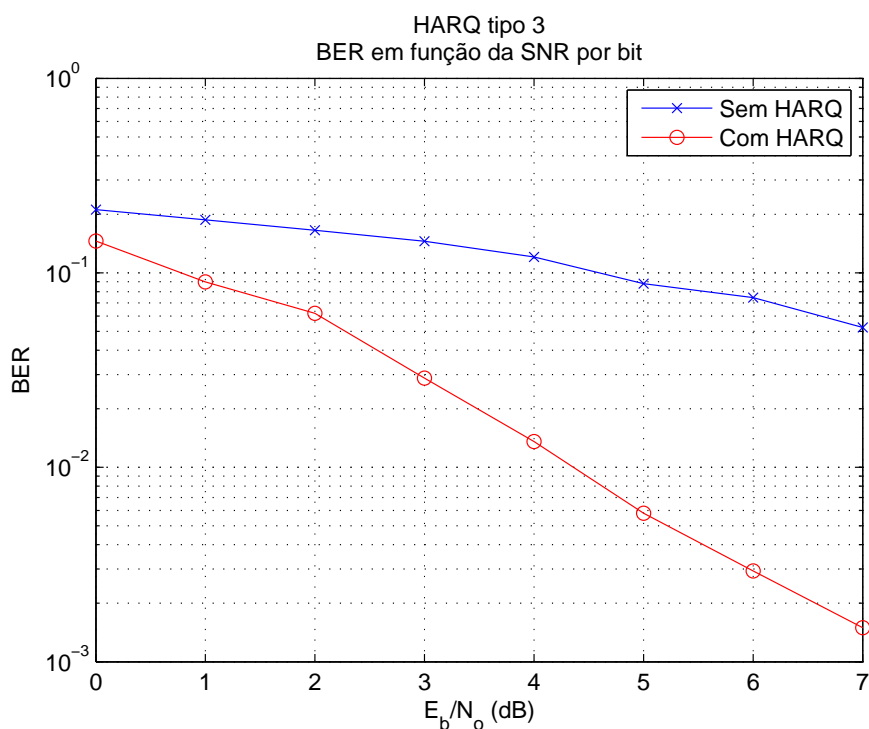


Figura 5.6: Ber em função da SNR por *bit* para 1 retransmissão (ou seja, sem HARQ) e 2 retransmissões HARQ tipo 3.

Assim como para o HARQ tipo 1, o HARQ tipo 3 melhora consideravelmente a BER para qualquer E_b/N_0 avaliadas. Novamente vê-se que o ganho fica mais expressivo com o aumento da E_b/N_0 .

A seguir, na Figura 5.7 avalia-se como a BER se altera com o aumento do limite máximo de retransmissão.

A cada retransmissão de pacotes, o decodificador possui cada vez mais informação acerca dos *bits* codificados, o que contribui para uma melhor decodificação. Assim como para o HARQ tipo 1, a cada retransmissão a BER melhora graças ao HARQ. Na terceira retransmissão, atingi-se erro inferior a 10^{-6} em $E_b/N_0 = 7 dB$, e na quarta retransmissão tem-se erro inferior a 10^{-6} a partir de $E_b/N_0 = 4 dB$. Os pontos de $BER < 10^{-6}$ não aparecem no gráfico.

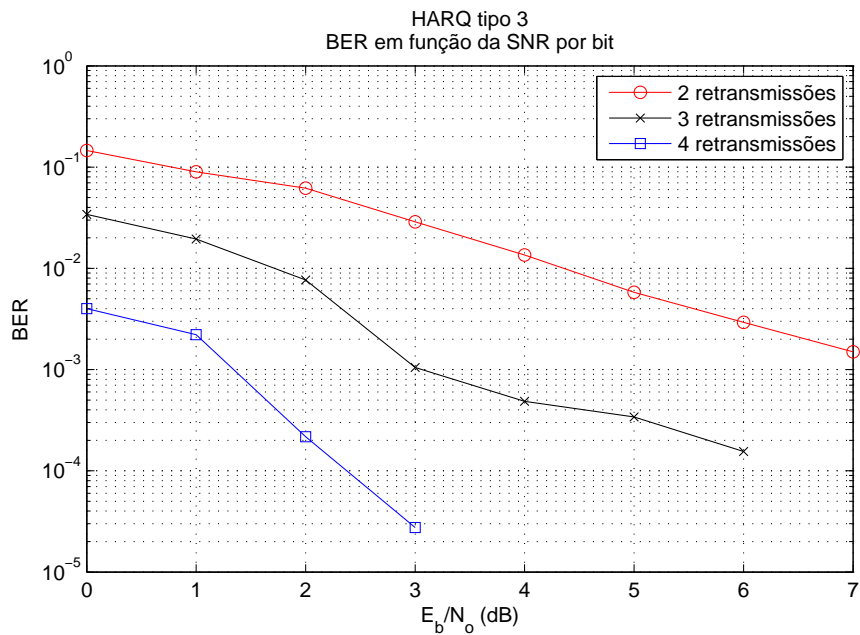


Figura 5.7: HARQ tipo 3 - BER em função da SNR por *bit* para sucessivas retransmissões.

5.2.2.3 Comparação entre HARQs

A Figura 5.8 contém as 3 curvas dos dois tipos de HARQ analisados.

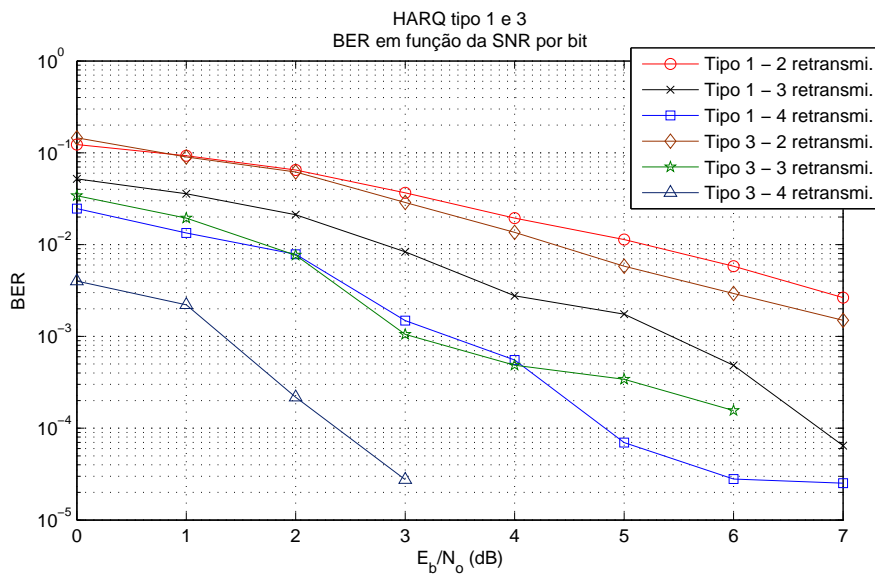


Figura 5.8: Comparação entre HARQ tipo 1 com *chase combining* e HARQ tipo 3.

Comparando as duas formas de HARQ, percebe-se que, para quase todas as SNRs por *bit* dos três pares de curvas analisadas, o HARQ tipo 3 possui taxa de *bits* errados menor que a taxa do HARQ tipo 1 com *Chase Combining*. Esse resultado reforça outros estudos feitos a respeito do tema [3]. Entretanto, é válido ressaltar que, para implementar o HARQ tipo 3 foi necessário ter espaço de armazenando para guardar cada pacote recebido em um *buffer* diferente, enquanto que

para implementar o HARQ tipo 1 com *Chase Combining* foi preciso ter espaço de armazenamento para guardar apenas um pacote. No padrão LTE utiliza-se HARQ tipo 3 sempre que possível, isto é, sempre que os terminais possuem espaço de armazenamento suficiente para guardar os pacotes recebidos. Entretanto, caso não seja possível empregar HARQ tipo 3, utiliza-se o HARQ tipo 1 com *Chase Combining*.

Capítulo 6

Conclusões

6.1 Considerações Finais

Neste trabalho, foi realizada a implementação do *rate matching* e do HARQ ao simulador WiSiL. Como foi explicado ao longo do trabalho, o *rate matching* e o HARQ são procedimentos importantes para atingir os objetivos do sistema LTE, isto é, oferecer alta taxa de transmissão de dados com alto grau de confiabilidade.

A variação da taxa de codificação permite que sejam utilizadas técnicas de modulação e codificação adaptativas, que consistem em adaptar o esquema de modulação e a taxa de codificação de acordo com a qualidade do canal de modo a obter as maiores taxas de transmissão de dados em um dado instante. Com isso, é possível usar melhor o canal e tornar o sistema mais eficiente espectralmente.

A simulação do HARQ se mostrou de grande valia como procedimento da camada MAC no padrão LTE. Houve melhoria significativa da taxa de *bits* errados para sucessivas retransmissões tanto para o HARQ tipo 1 com *Chase Combining*, como para o HARQ tipo 3. Também foi realizado um estudo comparativo entre os dois tipos de HARQ utilizados no padrão LTE e foi visto que o HARQ tipo 3, baseado na idéia de incremento de redundância, possui melhor desempenho que o HARQ tipo 1 com *Chase Combining*. Apesar disso, o desempenho do HARQ tipo 1 com *Chase Combining* foi bom o suficiente para ser implementado em situações onde não é possível utilizar o HARQ tipo 3.

Este trabalho deixa como legado algumas novas funções ao simulador WiSiL, que está agora um pouco mais completo.

6.2 Trabalhos Futuros

O simulador WiSiL ainda pode ser melhorado com a adição de novas funcionalidades para a simulação como:

- Implementação de modulação e codificação adaptativas;
- Otimização do código já feito;
- Simulação do enlace de subida do LTE utilizando SC-FDMA;
- Ampliação da simulação para englobar múltiplos usuários;
- Transmissão e recepção coordenada multi-ponto ou CoMP (*coordinated multi-point*);
- Simulação de candidatos a forma de onda da quinta geração.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ITU. *ICT facts and figures*. <http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2014-e.pdf>. Acessado em 20/11/2014.
- [2] GHOSH, A. et al. *Fundamentals of LTE*. [S.l.]: Prentice Hall, 2010.
- [3] KHAN, F. *LTE for 4G Mobile Broadband*. [S.l.]: Cambridge University Press, 2009.
- [4] REUTERS. *Qualcomm halts UMB project*. <http://www.reuters.com/article/2008/11/13/qualcomm-umb-idUSN1335969420081113?rpc=401&>. Acessado em 20/11/2014.
- [5] 3GPP. *3GPP TS 36.212 Multiplexing and channel coding*. Release 12. [S.l.], 2014.
- [6] PROAKIS, J. *Digital Communications*. [S.l.]: McGraw-Hill, 2007.
- [7] BERROU, C.; GLAVIEUX, A. Near-optimum error correcting coding and decoding: Turbo codes. *IEEE Transactions On Communications*, v. 44, n. 10, p. 1261–1271, October 1996.
- [8] FORNEY, D. Concatenated codes. *M.I.T. Press*, 1966.
- [9] SKLAR, B. *Digital Communications: Fundamentals and Applications*. [S.l.]: Prentice Hall, 2001.
- [10] MATHWORKS. *DL-SCH HARQ Modeling*. <http://www.mathworks.com/help/lte/examples/dl-sch-harq-modeling.html>. Acessado em 25/11/2014.
- [11] CHASE, D. Code combining - a maximum-likelihood decoding approach for combining an arbitrary number of noisy packets. *IEEE Transactions On Communications*, v. 33, p. 395–393, May 1985.
- [12] BUTT, M. M. Reliability-based packet combining with application to interleave-division multiple access. *6th International ITG-Conference on Source and Channel Coding*, 2006.
- [13] MORAIS, D. R. d. *Simulação da estimação de canal para um enlace direto de um sistema LTE usando múltiplas antenas*. Dissertao (Trabalho de conclusão de curso) — Universidade de Brasília, 2014.
- [14] 3GPP. *3GPP TR 25.996 Universal Mobile Telecommunications System (UMTS); Spatial channel model for Multiple Input Multiple Output (MIMO) simulations*. Release 12. [S.l.], 2014.