



Escuela de  
Ingeniería y Arquitectura  
**Universidad** Zaragoza

## PROYECTO FIN DE CARRERA

# Implementación y evaluación de las prestaciones de los códigos correctores de errores LDPC en un sistema de comunicaciones móviles WiMAX

**Raquel Sanz Segura**

Director:

**Jorge Ortín Gracia**

Ponente:

**Antonio Valdovinos**

Ingeniería de Telecomunicación

Especialidad Comunicaciones

Dpto. de Ingeniería Electrónica y Comunicaciones

Abril 2012



# Resumen

## **“Implementación y evaluación de las prestaciones de los códigos correctores de errores LDPC en un sistema de comunicaciones móviles WiMAX”**

Los códigos LDPC (Low-Density Parity-Check) son los códigos correctores de errores que presentan hasta la fecha unas prestaciones más cercanas a los límites teóricos establecidos por el teorema de Shannon, constituyendo uno de los mejores esquemas de codificación posibles. Por esta razón, esta clase de códigos se emplea en varios sistemas de comunicaciones de última generación, como por ejemplo en los sistemas de radiodifusión DVB-S2 (Digital Video Broadcasting - Satellite), DVB-C2 (Digital Video Broadcasting - Cable) y DVB-T2 (Digital Video Broadcasting - Terrestrial), este último usado en varios países para la transmisión de Televisión Digital Terrestre (TDT) de alta definición, en las redes fijas basadas en Ethernet a 10Gbps (estándar 802.3an), en redes LAN (Local Area Networks) inalámbricas (estándar 802.11n) o en las redes móviles basadas en WiMAX (Worldwide Interoperability for Microwave Access) (estándar 802.16e).

El objetivo del proyecto es implementar un sistema de codificación LDPC y evaluar sus prestaciones en un sistema de comunicaciones real. Para ello, se emplearán los códigos LDPC especificados en el estándar 802.16e y se integrarán en un simulador de capa física del sistema WiMAX que permitirá evaluar sus prestaciones en un entorno de comunicaciones móviles. Esto requerirá programar en C++ tanto el bloque de codificación en el emisor como el de decodificación en el receptor. Asimismo, se ha de modificar la interfaz de usuario del simulador para permitir la elección de determinados parámetros del codificador tales como la tasa de codificación o el tamaño de bloque FEC (Forward Error Correction).

La principal característica de los códigos LDPC consiste en que sus matrices de chequeo de paridad contienen sólo unos pocos 1's en comparación con la cantidad de 0's que presentan, por lo que el proceso de codificación y decodificación es muy eficiente. Para realizar la decodificación se implementará el algoritmo suma-producto y una simplificación del mismo. Este algoritmo se basa en el cálculo iterativo de las estimaciones de los distintos bits codificados dado el vector de datos recibidos y la estructura inherente del código, que fuerza a que el síndrome del vector de datos decodificados sea igual a 0.

Finalmente, se evaluarán las prestaciones de los códigos LDPC en términos de BER (Bit Error Rate) y BLER (Block Error Rate) para diferentes SNR y condiciones de canal, así como la carga computacional requerida para codificar y decodificar. En este sentido, se evaluará la influencia del tamaño de bloque y la tasa de transmisión tanto en las prestaciones de los códigos como en su complejidad para ser decodificados.



# Índice general

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Contexto . . . . .	1
1.2	Objetivos del proyecto . . . . .	2
1.3	Estructura de la memoria . . . . .	3
<b>2</b>	<b>Estado del arte</b>	<b>5</b>
2.1	WiMAX . . . . .	5
2.2	Introducción a los sistemas digitales de Telecomunicaciones . . . . .	8
2.2.1	Los códigos bloque . . . . .	10
2.2.2	Códigos LDPC . . . . .	10
<b>3</b>	<b>Los códigos LDPC</b>	<b>13</b>
3.1	Introducción a los códigos LDPC . . . . .	13
3.1.1	Representación de los códigos LDPC . . . . .	13
3.1.1.1	Representación matricial . . . . .	14
3.1.1.2	Representación gráfica . . . . .	14
3.1.2	Diferentes formas sistemáticas de un código bloque . . . . .	15
3.1.3	Descripción de los códigos LDPC . . . . .	16
3.1.4	Construcción de los códigos LDPC . . . . .	17
3.1.4.1	Códigos regulares . . . . .	17
3.1.4.2	Códigos irregulares . . . . .	18
3.2	Codificación LDPC . . . . .	18
3.2.1	Generación de la matriz de paridad . . . . .	18
3.2.2	Codificación LDPC . . . . .	21
3.3	Decodificación LDPC . . . . .	22

3.3.1	Decodificación de los códigos LDPC: El grafo de Tanner . . . . .	22
3.3.2	Algoritmo suma-producto simplificado . . . . .	24
<b>4</b>	<b>Arquitectura del sistema WiMAX</b>	<b>27</b>
4.1	Descripción general del escenario utilizado . . . . .	27
4.1.1	Modulador . . . . .	28
4.1.2	Transmisor básico . . . . .	29
4.1.3	Canal: Tipo y parámetros . . . . .	30
4.1.4	Receptor ZF . . . . .	31
4.1.5	Demodulador . . . . .	31
<b>5</b>	<b>Propuesta de codificador/decodificador bloque para sistemas WiMAX: el código LDPC</b>	<b>33</b>
5.1	El codificador bloque LDPC . . . . .	34
5.1.1	Randomizador . . . . .	34
5.1.2	El codificador . . . . .	35
5.2	El decodificador bloque LDPC . . . . .	35
<b>6</b>	<b>Simulación y análisis de resultados</b>	<b>39</b>
6.1	Consideraciones generales de las simulaciones . . . . .	39
6.2	Efecto del tamaño de bloque FEC en la decodificación LDPC . . . . .	40
6.3	Efecto del tipo de modulación y de la tasa de código . . . . .	42
6.4	Efecto del número máximo de iteraciones permitidas en la decodificación .	44
6.5	Análisis de la carga computacional del decodificador LDPC . . . . .	47
<b>7</b>	<b>Conclusiones</b>	<b>49</b>
7.1	Conclusiones . . . . .	49
	<b>Bibliografía</b>	<b>51</b>
<b>A</b>	<b>Acrónimos</b>	<b>53</b>

# Índice de figuras

2.1	Arquitectura del sistema WiMAX de acceso móvil (estándar 802.16e) . . .	6
2.2	Diagrama básico de un sistema de comunicaciones digitales . . . . .	8
3.1	Grafo de Tanner correspondiente a la matriz de chequeo de paridad representada en 3.1 . . . . .	14
3.2	Matriz modelo $H_{mb}$ para cada ratio de código . . . . .	20
3.3	Grafo de Tanner. Grafo bipartito que une los nodos símbolo con los nodos de chequeo de paridad . . . . .	23
4.1	Diagrama del sistema de comunicaciones WiMAX utilizado . . . . .	28
4.2	Generación de un símbolo OFDM . . . . .	29
5.1	Diagrama de bloque del proceso de codificación estipulado por el estándar WiMAX . . . . .	33
5.2	Diagrama de bloque del proceso de codificación propuesto para sistemas WiMAX . . . . .	34
5.3	Generador PRBS para el proceso de aleatorización en WiMAX . . . . .	34
5.4	Proceso de formación de las secuencias de aleatorización para el randomizador en el enlace descendente (arriba) y el ascendente (abajo) . .	35
6.1	BLER para QPSK, tasa 1/2 y canal AWGN . . . . .	40
6.2	BLER para QPSK, tasa 1/2 y canal Pedestrian A . . . . .	41
6.3	BLER para QPSK, tasa 1/2 y canal Vehicular A . . . . .	41
6.4	BLER para QPSK, FEC fijo y canal AWGN . . . . .	42
6.5	BLER para QPSK, FEC fijo y canal Pedestrian A . . . . .	42
6.6	BLER para QPSK, FEC fijo y canal Vehicular A . . . . .	43
6.7	BLER para 16QAM, FEC fijo y canal AWGN . . . . .	43
6.8	BLER para 16QAM, FEC fijo y canal Pedestrian A . . . . .	43

6.9	BLER para 16QAM, FEC fijo y canal Vehicular A . . . . .	43
6.10	BLER para QPSK, FEC fijo y canal AWGN. Efecto del número máximo de iteraciones . . . . .	45
6.11	BLER para QPSK, FEC fijo y canal Pedestrian A. Efecto del número máximo de iteraciones . . . . .	45
6.12	BLER para QPSK, FEC fijo y canal Vehicular A. Efecto del número máximo de iteraciones . . . . .	45
6.13	Canal AWGN, número medio de iteraciones en función de la SNR . . . . .	46
6.14	Canal Pedestrian A, número medio de iteraciones en función de la SNR . . . . .	46
6.15	Canal Vehicular A, número medio de iteraciones en función de la SNR . . . . .	46



# Índice de Tablas

4.1	Valores seleccionados para cada canal . . . . .	30
6.1	Parámetros de la simulación para el estudio del efecto del tamaño de bloque FEC . . . . .	40
6.2	Parámetros de la simulación para el estudio del efecto de la tasa de código	42
6.3	Parámetros de la simulación para el estudio del efecto del número máximo de iteraciones permitido . . . . .	44
6.4	Número de operaciones necesarias para la decodificación de un bloque FEC utilizando el decodificador LDPC . . . . .	47



# Capítulo 1

## Introducción

### 1.1 Contexto

A lo largo de los últimos años los hábitos de comunicación de la sociedad están cambiando, principalmente por el aumento de movilidad de las personas. Esta movilidad ha hecho que los sistemas de comunicaciones móviles 3G y 4G cobren una gran relevancia. Puesto que la tecnología 3G ha alcanzado ya todo su potencial tecnológico y de negocio, está empezando a cobrar especial importancia el concepto de 4G (también llamada B3G, *Beyond 3G*); que se ha concebido como una evolución convergente de la 3G basada en la combinación de múltiples redes, tecnologías móviles y de acceso inalámbrico que completan a los actuales sistemas 3G.

La tecnología 4G está totalmente basada en protocolo IP, lo que la convierte en un sistema de sistemas y una red de redes; y está previsto que pueda alcanzar velocidades de acceso entre 100 Mbps en movimiento y 1 Gbps en reposo, manteniendo una calidad de servicio (QoS) de punta a punta (*end-to-end*) de alta seguridad para permitir ofrecer servicios de cualquier clase en cualquier momento, en cualquier lugar, con el mínimo coste posible.

El WWRF (Wireless World Research Forum) define 4G como una red que funcione en la tecnología de Internet, combinándola con otros usos y tecnologías tales como Wi-Fi (Wireless-Fidelity) y WiMAX (World Wide Interoperability for Microwave Access). La 4G no es una tecnología o estándar definido, sino una colección de tecnologías y protocolos para permitir el máximo rendimiento de procesamiento con la red inalámbrica más barata.

En la actualidad, existen dos grandes tecnologías o sistemas que ya utilizan la tecnología móvil 4G. Uno de ellos es el sistema LTE (Long Term Evolution); mientras que el segundo es el conocido como WiMAX, siendo este último el sistema de tecnología móvil en el que se engloba el estudio de este proyecto.

Debido a la necesidad de obtener unas elevadas tasas de transmisión en anchos de banda limitados, en estos sistemas cobra una especial relevancia los esquemas de

codificación de canal empleados para mejorar la tasa de error en el bit (Bit Error Rate - BER) y aumentar de este modo la eficiencia del sistema en términos de la tasa de transmisión de información útil por unidad de ancho de banda que se puede alcanzar.

Los códigos LDPC (Low-Density Parity-Check) son los códigos correctores de errores que presentan hasta la fecha unas prestaciones más cercanas a los límites teóricos establecidos en el teorema de Shannon, siendo por tanto superiores a otros esquemas de codificación. Esta clase de códigos se emplea en varios sistemas de comunicaciones de última generación, como por ejemplo en los sistemas de radiodifusión basados en los estándares DVB-S2 y DVB-T2 obteniendo grandes resultados en términos de BER y BLER (Block Error Rate) en función de la SNR (Signal to Noise Ratio).

Por esta razón, esta clase de códigos van a ser el objeto de estudio de este proyecto; implementándolos en un sistema de redes móviles WiMAX (estándar 802.16e) y estudiando la clase de prestaciones que aportan en estos sistemas de última generación.

## 1.2 Objetivos del proyecto

Los principales objetivos que se han perseguido durante la realización de este proyecto son:

- Realizar un estudio de los códigos bloque LDPC para conocer su método de funcionamiento y de los algoritmos de codificación y decodificación; así como de las características que especifica el estándar WiMAX en cuanto a estos bloques para su implantación en dicho sistema.
- Implementar un sistema de codificación LDPC (especificado en el estándar 802.16e) e integrarlo en un simulador de capa física del sistema WiMAX.
- Realizar un estudio completo en cuanto a tasas de código, tamaño de los bloques, etc., a utilizar y evaluar las prestaciones que se consiguen en términos de probabilidad de error en el bloque (BLER). Estos estudios, con el fin de comparar los resultados, se realizarán a partir de resultados obtenidos mediante simulación en diferentes canales: canal gaussiano ideal, y canales móviles más realistas.
- Realizar un estudio del efecto del número máximo de iteraciones permitido en el algoritmo de decodificación iterativo en la BLER. Este estudio se realizará también mediante la obtención de resultados numéricos simulados en el sistema WiMAX de capa física y en diferentes canales móviles.
- Evaluación del coste computacional requerido en la decodificación.

Para la evaluación de las prestaciones en un entorno de comunicaciones móviles, los algoritmos que implementan al codificador y decodificador (y de los que se obtienen los resultados numéricos) han sido programados en el lenguaje de programación C++ e introducidos en un simulador de capa física del sistema WiMAX. Este simulador incluye

todos los bloques propios de un sistema básico de comunicaciones WiMAX, como son el transmisor, canal móvil y receptor, tal y como se definen en los estándares IEEE 802.16e; a los que se ha añadido dos nuevos bloques encargados de la codificación LDPC en el transmisor y la decodificación LDPC en el receptor. Todo este sistema completo, va a permitir simular y analizar resultados de las prestaciones que estos codificadores aportan a los sistemas de comunicaciones móviles WiMAX.

### 1.3 Estructura de la memoria

A continuación va a realizarse una pequeña descripción de los capítulos de que consta esta memoria:

- **Capítulo 1:** Este capítulo realiza una breve introducción al contexto en el que se va a englobar este proyecto, haciendo una mención al sistema de comunicaciones móviles que va a ser utilizado (WiMAX). Asimismo, se detallan los objetivos que persigue la realización de este proyecto fin de carrera y la estructura que va a seguir esta memoria.
- **Capítulo 2:** Este capítulo, titulado *Estado del arte*, pretende dar una visión general de los aspectos relevantes para la comprensión de este proyecto, de modo que va a darse una visión general de las características del sistema de comunicaciones móviles WiMAX. Por otro lado, se va a realizar una introducción matemática al teorema de Shannon así como una breve introducción a la codificación bloque LDPC, que es la que va a usarse en el proyecto.
- **Capítulo 3:** Va a realizarse una descripción completa que permita la total comprensión de los códigos LDPC así como de su método de codificación y decodificación.
- **Capítulo 4:** En este capítulo describiremos los bloques básicos que componen la capa física de un sistema WiMAX, así como de los parámetros que se especifican para cada bloque según el estándar WiMAX. Se explicarán en más detalle los bloques introducidos al sistema WiMAX del que se partía y que son el objeto de estudio de este proyecto (codificador y decodificador LDPC).
- **Capítulo 5:** Realizaremos una descripción de los distintos estudios llevados a cabo y se mostrarán los resultados de las simulaciones de dichos estudios gráficamente en términos de BLER conseguidos en función de la SNR. Por último se realizará un breve estudio del coste computacional que estos códigos añaden.
- **Capítulo 6:** Se detalla una recopilación de las conclusiones extraídas de los resultados obtenidos y se proponen posibles líneas futuras de continuación al trabajo realizado en este PFC.

Por último, se añade a esta memoria un apartado de **Bibliografía** en el que se detallan todos los libros, artículos, etc. a los que se hacen referencia en esta memoria; y

un apartado de ***Acrónimos*** que lista todas las siglas utilizadas en la redacción de esta memoria así como su significado.

## Capítulo 2

# Estado del arte

### 2.1 WiMAX

WiMAX, como ya se ha dicho en el capítulo 1, es un sistema de comunicaciones inalámbricas de última milla, por lo que permite el acceso de los usuarios a distintos servicios de comunicaciones como puede ser telefonía o conectividad a Internet. El estándar en el que está basada esta tecnología es el IEEE 802.16. Una de sus principales ventajas es dar servicios de banda ancha en zonas donde el despliegue de cable o fibra por la baja densidad de población presenta unos costos por usuario muy elevados, por lo que es principalmente atractivo para zonas rurales.

WiMAX está diseñado como una alternativa wireless al acceso de banda ancha DSL (Digital Subscriber Line) y cable, y una forma de conectar nodos Wi-Fi en una red de área metropolitana (MAN). WiMAX puede proveer de acceso de banda ancha wireless de hasta 80 kilómetros, y algunas de las ventajas de este sistema son:

- Puede dar cobertura a un área bastante extensa y la instalación de las antenas (estaciones base) es rápida y sencilla. Esto lo hace adecuado para dar cobertura a ciudades enteras, pudiendo formar una MAN, en lugar de un área de red local como puede proporcionar Wi-Fi.
- WiMAX tiene una velocidad de transmisión mayor que la de Wi-Fi, pudiendo alcanzar los 75 Mbps (35 + 35 Mbps) para WiMAX de acceso fijo (estándar 802.16d) o de 30 Mbps (ancho de banda de 10 MHz) para acceso móvil (estándar 802.16e), siempre y cuando el espectro esté totalmente limpio.
- Puede ser simétrico, lo que significa que puede proporcionar un flujo de datos similar tanto de subida como de bajada.
- Facilidades para añadir más canales, dependiendo de la regulación de cada país.
- Anchos de banda configurables y no cerrados, sujeto a la relación de espectro.

Como ya se ha dicho anteriormente, el estándar IEEE 802.16 forma las bases del sistema

WiMAX. Existen dos variantes dentro del estándar IEEE 802.16, una de acceso fijo (802.16d) y otra de movilidad completa (802.16e). En cuanto a la localización del espectro para WiMAX, no hay uniformidad global en las licencias de uso; sin embargo, el WWFR ha publicado tres perfiles de espectro con licencia: 2.3 GHz, 2.5 GHz y 3.5 GHz, en un esfuerzo por impulsar la estandarización de este sistema y su menor costo.

Por las especiales características en cuanto al ancho de banda y el rango de acción de WiMAX, lo hacen especialmente interesantes para aplicaciones como:

- Proporcionar conectividad de banda ancha móvil a través de ciudades y países y para una gran variedad de dispositivos.
- Proporcionar una alternativa sin cables para el acceso de banda ancha en la última milla.
- Proporcionar servicios de datos, telecomunicaciones (Voice over IP - VoIP) e IPTV.
- Proporcionar una fuente de conexión a Internet como parte de un plan de continuidad del negocio (menor probabilidad de interrupción del servicio en las empresas).

El objeto de estudio de este proyecto está basado en el estándar IEEE 802.16e, que permite el desplazamiento del usuario de un modo similar al que se puede dar en GSM/UMTS y actualmente es el que compite con las tecnologías LTE por ser la alternativa para las operadoras de telecomunicaciones que apuestan por los servicios de movilidad. WiMAX 802.16e es capaz de ofrecer 30 Mbps (ancho de banda de 10 MHz) en la banda frecuencial de 2-6 GHz en celdas de hasta 5 Km de radio.

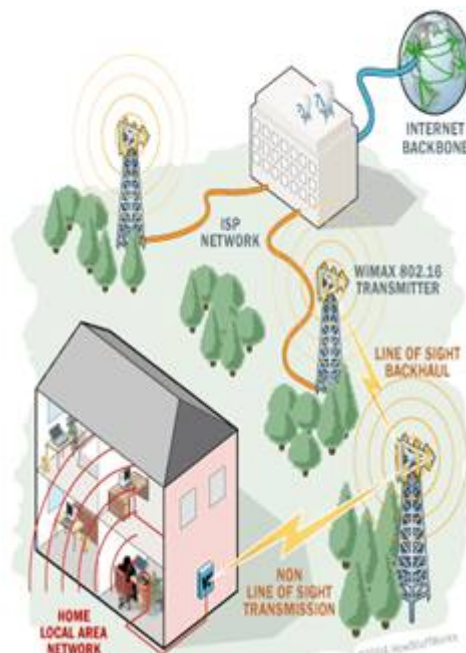


Figura 2.1: Arquitectura del sistema WiMAX de acceso móvil (estándar 802.16e)



Puesto que el estándar 802.16e ofrece un enorme añadido sobre la versión fija por su movilidad (ya que además ofrece un opción de conexión fija, como puede observarse en la Figura 2.1) es esta alternativa la que se está imponiendo a WiMAX de acceso fijo; quedando esta última relegada a conexiones *backhaul* usadas para interconectar la red central (*backbone*) con las diferentes subredes usando diferentes tipos de tecnologías alámbricas o inalámbricas. Con el fin de conseguir altas tasas de transmisión con costes mínimos, ampliación y mejora de los servicios y una mayor integración con los protocolos ya existentes, las tecnologías 4G (WiMAX y LTE) han introducido nuevos avances tecnológicos tales como:

- MIMO (Multiple-Input Multiple-Output), permite aumentar la eficiencia espectral del sistema de comunicación inalámbrica por medio de la utilización de varias antenas tanto en el transmisor como en el receptor, aprovechando así el fenómeno de la propagación multicamino.
- OFDM (Orthogonal Frequency Division Multiplexing), que permite aumentar la eficiencia espectral y combatir las interferencias multicamino con gran robustez. El principio básico de funcionamiento de esta técnica consiste en enviar un conjunto de portadoras a diferentes frecuencias, donde cada portadora transporta la información previamente modulada en PSK o QAM. En la mayoría de los sistemas, la multiplexación OFDM se realiza tras pasar la información por un codificador de canal con el objetivo de corregir errores producidos en la transmisión. Este tipo de multiplexación es denominado como COFDM (*Coded OFDM*). Además existe una versión multiusuario conocida como OFDMA (Orthogonal Frequency Division Multiple Access) que es la más ampliamente utilizada en sistemas 4G (añadiendo la codificación de canal) y que permite compartir el espectro a través de la división del ancho de banda en un conjunto de subportadoras que se reparten en grupos en función de la necesidad de cada usuario [1].

Si bien OFDM presenta múltiples ventajas frente a las modulaciones que emplean una única portadora (ecualización en el receptor más sencilla, robustez frente a propagación multicamino, alta eficiencia espectral, implementación eficiente mediante FFT, simplicidad de la ecualización en recepción), también presenta una serie de vulnerabilidades que degradan su funcionamiento en entornos inalámbricos móviles. En esta clase de sistemas, es habitual la pérdida de la información contenida en parte de las portadoras al transmitirse en canales con desvanecimientos, lo que hace necesario el empleo de estrategias de corrección que recuperen los errores introducidos en aquellas portadoras atenuadas. Entre estas estrategias destacan los códigos LDPC, los cuales consiguen valores de capacidad cercanos a los límites establecidos por el teorema de Shannon, cuyo estudio constituye el objeto del presente proyecto.

## 2.2 Introducción a los sistemas digitales de Telecomunicaciones

El objetivo de todo diseñador de sistemas digitales de telecomunicaciones (cuyo esquema básico es mostrado en la figura 2.2) es conseguir superar los problemas que supone la transmisión de datos en un entorno ruidoso, proporcionando así una comunicación con unos niveles de fiabilidad y velocidad suficientemente buenos para una aplicación concreta. Dos de los parámetros fundamentales con los que puede jugar el ingeniero son el ancho de banda disponible  $W$  y la potencia transmitida  $S$ . El ancho de banda  $W$  es un bien escaso en el mundo de las telecomunicaciones, y por otro lado, la potencia transmitida  $S$  tampoco puede ser ilimitada. Ambos factores, junto con la densidad espectral de potencia de ruido en el receptor, fijan para cada modulación la relación señal a ruido por bit, así:

$$\frac{E_b}{N_0} = \frac{ST_b}{N_0} = \frac{SW}{NR_b} \quad (2.1)$$

donde  $N$  es la potencia de ruido en recepción (medida en vatios),  $T_b$  es el tiempo de duración de cada bit (medido en segundos) y  $R_b$  es la velocidad de transmisión (en bits/segundo), parámetro directamente relacionado con el ancho de banda  $W$  (Hz). La importancia del valor  $E_b/N_0$  es que éste determina la probabilidad de error en el bit  $P_b$  para un esquema de modulación establecido.

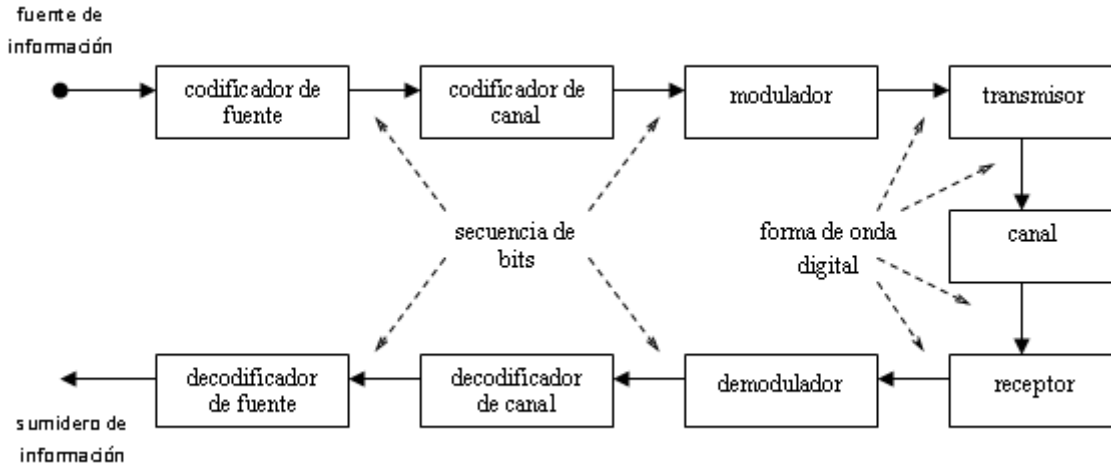


Figura 2.2: Diagrama básico de un sistema de comunicaciones digitales

El famoso teorema de Shannon-Hartley [2] determina un límite para la capacidad de transmisión de información  $C$ , en función del ancho de banda  $W$  y la relación señal a ruido  $S/N$ , que es

$$C = W \log_2 \left( 1 + \frac{S}{N} \right) \quad (2.2)$$

Este teorema (ecuación 2.2) nos viene a decir que existe una posibilidad teórica de transmitir a cualquier tasa  $R_b < C$  con una probabilidad de error arbitrariamente pequeña si usamos un sistema de codificación adecuado. Además, se ha demostrado [3] que empleando una serie de transformaciones matemáticas, el teorema de Shannon-Hartley puede ser visto como

$$1 = \frac{E_b}{N_0} \log_2(1 + x)^{(1/x)} \quad (2.3)$$

con

$$x = \frac{E_b C}{N_0 W} \quad (2.4)$$

En numerosas ocasiones el diseñador del sistema se encuentra con que es incapaz de alcanzar en primera instancia una probabilidad de error  $P_b$  suficientemente baja con los parámetros de los que dispone. De forma que si quiere mejorar su curva  $(E_b/N_0, P_b)$  y acercarla en algunos dBs hacia el límite de Shannon debe rebajar el número de errores en la comunicación, y la única opción práctica es entonces el uso de codificación de canal. Otra forma de valorar lo acertado de esta decisión consistiría en darnos cuenta de que, con codificación de canal, el diseñador podrá ahorrar energía transmitida y mantener por otro lado la probabilidad de error que tenía sin codificación de canal (pero que conseguía gastando más potencia de señal).

En general, la codificación de canal se refiere a un vasto conjunto de posibles modificaciones a la señal para hacerla más robusta ante los efectos degradantes del canal (ruido, fading, etc...), de forma que podamos reducir la probabilidad de error o bien la relación señal a ruido necesaria para obtener una tasa de error conveniente. El precio a pagar es generalmente un aumento en el ancho de banda, excepto en sistemas que combinen modulación y codificación como el TCM (Trellis Coded Modulation).

Sklar [3] distinguía entre “Codificación de Formas de Onda” (*Waveform Coding*) y “Secuencias Estructuradas” (*Structured Sequences*) como las dos áreas principales a investigar en la teoría de codificación de canal. El primer grupo estudia modificar las propias formas de onda para conseguir una mejor decodificación, exenta de errores en la medida de lo posible. Es poco frecuente, y no vamos a dar más detalles acerca de él. Centraremos nuestro estudio en el segundo grupo, secuencias estructuradas, empleado masivamente en la práctica totalidad de los sistemas de comunicaciones actuales. Básicamente busca algoritmos matemáticos para modificar las secuencias binarias antes de la modulación e introducir bits de redundancia, de forma que la información viaje más protegida.

Existen dos formas de hacer frente a los errores en la transmisión cuando empleamos redundancia. La primera de ellas es la llamada *Detección de Errores y Retransmisión* o ARQ (Automatic Repeat Request) donde los bits de redundancia se usan para verificar si ha habido error o no, pidiendo la retransmisión de los datos en caso necesario; por lo que dichos bits de redundancia no se usan directamente para corregir los errores. La segunda estrategia se denomina FEC (Forward Error Correction), y en esta estrategia se incluyen los códigos bloque (y en particular los códigos bloque LDPC que van a ser

estudio de este proyecto) y en general cualquier sistema de codificación que emplee la información redundante para corregir errores.

A continuación explicaremos brevemente el principio básico de funcionamiento de un tipo de codificadores de canal: los codificadores bloque. También introduciremos brevemente un caso especial de estos códigos bloque: los códigos LDPC, que van a ser el objeto de estudio de este proyecto.

### 2.2.1 Los códigos bloque

En una codificación bloque, a cada mensaje de entrada de  $k$  bits se le asigna un código de salida de  $n$  bits (con  $n > k$ ). El bloque encargado de realizar esta asignación se denomina codificador de canal y al conjunto de  $2^k$  palabras se le denomina *código de canal*. El principio que se utiliza en los códigos bloque consiste en estructurar los datos en bloques de longitud fija y añadir a cada bloque un cierto número de bits llamados bits de redundancia. Sólo ciertas combinaciones de bits son aceptables y forman una colección de palabras de código válidas. A la hora de obtener el bloque original en el receptor, los bits de redundancia pueden ser utilizados para corregir los errores que el canal haya podido introducir.

La característica más destacada de los códigos bloque es que cada bloque de  $n$  bits o palabra código generada en el codificador depende solamente del correspondiente bloque de  $k$  bits generado por la fuente de información, siendo por lo tanto una *codificación sin memoria*.

Los códigos bloque pueden ser lineales o no lineales. Un código lineal se define mediante una asignación lineal del espacio de mensajes de entrada al espacio de palabras código, y puede representarse como un producto de matrices. Los códigos lineales se denominan también *códigos de comprobación de paridad*, pues la palabra código se obtiene a partir de sumas módulo dos de subconjuntos de los bits de entrada. Un código de este tipo queda completamente caracterizado por una *matriz generadora*  $G$ .

### 2.2.2 Códigos LDPC

Los códigos LDPC son una clase especial de códigos bloque lineales. El nombre de estos códigos viene de las características que presenta la matriz de comprobación de paridad, la cual contiene sólo unos pocos unos en comparación con la cantidad de ceros. Si el número de unos es fijo para cada fila y para cada columna, estos códigos son llamados *códigos LDPC regulares*. La principal característica de este tipo de codificadores, reside en que la codificación y decodificación se realiza a partir de la matriz  $H$ , donde cada fila indica una ecuación de paridad y cada columna es cada uno de los símbolos presentes en la transmisión; así, cuando un 1 está presente en una de las ecuaciones de paridad, indica que el símbolo  $j$ , indicado por la columna en la que se encuentra, interviene en esa ecuación de paridad.

Estos códigos fueron introducidos por primera vez por Gallager en 1960 [4]. Pero,

debido al gran esfuerzo computacional que se requería en la implementación del codificador y el decodificador y a la introducción de los códigos Reed-Solomon, los códigos LDPC fueron ignorados hasta hace unos diez años.

Los códigos LDPC han sido estudiados en profundidad en los últimos años y se han hecho grandes progresos en la comprensión y en la habilidad de diseñar sistemas de codificación iterativos. La propuesta de decodificación iterativa es ya usada en turbo códigos, pero la estructura de los códigos LDPC dan incluso mejores resultados. En muchos casos, permiten una tasa de codificación mayor y también un ratio de error menor. Además hacen posible la implementación de decodificadores paralelizables. Las principales desventajas de estos códigos, es que los codificadores son, en algunas ocasiones, más complejos y que la longitud del código tiene que ser bastante larga para que de buenos resultados.



## Capítulo 3

# Los códigos LDPC

### 3.1 Introducción a los códigos LDPC

En su artículo de 1948 [2], Shannon estableció los límites teóricos para el comportamiento de la codificación de corrección de errores. Desde entonces, muchos esquemas de corrección de errores han sido propuestos, pero ninguno ha logrado alcanzar comportamientos cercanos al ideal hasta que no se propusieron los esquemas de turbo codificación, descubiertos en 1993 por Berrou, Glavieux y Thitimajshima [5]. Tres años después, en 1996, Mackay y Neal [6][7] redescubrieron una clase de códigos introducidos inicialmente por Gallager en 1960 [4] que han conseguido también alcanzar un comportamiento cercano al ideal. Estos códigos de Gallager van a ser el objeto de estudio de este proyecto, así como su codificación y decodificación.

Los códigos Gallager, o más comúnmente conocidos como códigos LDPC, son códigos bloque lineales contruidos a partir del diseño de una matriz  $H$  de chequeo de paridad poco densa. Esto es, para el caso binario, una matriz que contenga unos pocos 1's en comparación con la cantidad de 0's presentes en dicha matriz. Gallager, en su tesis doctoral, también presentó un método iterativo para la decodificación de este tipo de códigos, los cuales fueron capaces de lograr excelentes resultados. Sin embargo, la complejidad de estos algoritmos iterativos de decodificación estaba más allá de las capacidades de los procesadores de entonces, lo que hizo que estos códigos fueran olvidados hasta 1996.

#### 3.1.1 Representación de los códigos LDPC

Básicamente, hay dos posibilidades diferentes de representar códigos LDPC. Como todos los códigos bloque lineales, pueden ser descritos de forma matricial. La segunda posibilidad es mediante la representación gráfica. A continuación vamos a realizar una breve descripción de ambos métodos de representación que serán más desarrollados en secciones posteriores.

### 3.1.1.1 Representación matricial

Podemos definir dos números para describir este tipo de matrices (matrices de comprobación de paridad de baja densidad). Éstos números son:  $v$ , que representa el número de unos por fila y,  $s$  para el número de unos por columna. Para que una matriz pueda ser llamada de baja densidad, se deben cumplir dos condiciones:  $s \ll n$  y  $v \ll m$ , donde  $m \times n$  es la dimensión de la matriz con  $n$  el número de columnas y  $m$  el número de filas. Para poder cumplir estas condiciones, la matriz de chequeo de paridad debe ser muy grande.

### 3.1.1.2 Representación gráfica

Tanner [8] introdujo una forma efectiva de representar gráficamente los códigos LDPC: *el grafo de Tanner*. Este gráfico no sólo provee una completa representación gráfica del código, sino que también ayuda a describir el algoritmo de decodificación que será explicado más adelante.

En los grafos de Tanner hay dos conjuntos distintos de nodos, pudiendo haber únicamente arcos entre nodos pertenecientes a conjuntos diferentes. Los dos tipos diferentes de nodos en los grafos de Tanner son llamados *nodos de símbolos*  $d_j$  (que corresponden a cada bit del vector codificado) y *nodos de chequeo de paridad*  $h_i$  (correspondientes a cada ecuación de chequeo de paridad). Cada arco en el gráfico de Tanner representa la participación de ese bit del vector codificado en el cálculo de la ecuación de paridad a la cual está unida; así, cada ecuación de paridad queda definida por todos aquellos bits del vector codificado (nodos símbolo) a los cuales está unida.

$$H = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix} \quad (3.1)$$

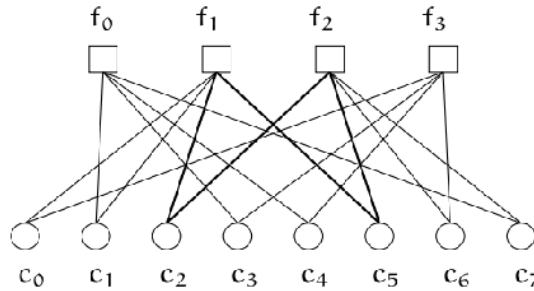


Figura 3.1: Grafo de Tanner correspondiente a la matriz de chequeo de paridad representada en 3.1



### 3.1.2 Diferentes formas sistemáticas de un código bloque

Un código bloque  $C_b(n, k)$  se puede especificar completamente empleando su matriz generadora, la cual, en el caso de códigos bloque sistemáticos, es de la forma:

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0,n-k-1} & 1 & 0 & 0 & \cdots & 0 \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (3.2)$$

que se puede expresar con la notación:

$$\mathbf{G} = [\mathbf{P} \mathbf{I}_k] \quad (3.3)$$

donde  $\mathbf{P}$  es la submatriz de paridad e  $\mathbf{I}_k$  es la submatriz de identidad de dimensión  $k \times k$ . En esta forma de codificación sistemática, los bits del mensaje original aparecen al final del vector código.

Para realizar la codificación se multiplica cada vector de datos  $\mathbf{m}$ , de dimensión  $k \times 1$ , por la traspuesta de la matriz generadora  $\mathbf{G}$ , de dimensión  $k \times n$ , obteniéndose el vector de datos codificados  $\mathbf{c}$ , también denominado vector código, de dimensión  $n \times 1$ :

$$\mathbf{c} = \mathbf{G}^T \circ \mathbf{m} \quad (3.4)$$

La forma sistemática de la matriz de chequeo de paridad  $\mathbf{H}$  del código  $C_b$  generada por la matriz generadora  $\mathbf{G}$  es:

$$H = \begin{bmatrix} 1 & 0 & \cdots & 0 & p_{00} & p_{10} & \cdots & p_{k-1,0} \\ 0 & 1 & \cdots & 0 & p_{01} & p_{11} & \cdots & p_{k-1,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix} = [\mathbf{I}_{n-k} \mathbf{P}^T] \quad (3.5)$$

donde  $\mathbf{P}^T$  es la traspuesta de la matriz de paridad  $\mathbf{P}$ . La matriz de chequeo de paridad es tal que el producto entre un vector fila  $\mathbf{g}_i$  de la matriz generadora  $\mathbf{G}$  y un vector fila  $\mathbf{h}_j$  de la matriz de chequeo de paridad  $\mathbf{H}$  es cero; esto es,  $\mathbf{g}_i$  y  $\mathbf{h}_j$  son ortogonales. Por tanto,

$$\mathbf{H} \circ \mathbf{G}^T = \mathbf{0} \quad (3.6)$$

y entonces

$$\mathbf{H} \circ \mathbf{c} = \mathbf{H} \circ \mathbf{G}^T \circ \mathbf{m} = \mathbf{0} \quad (3.7)$$

Como veremos más adelante, el diseño de un código LDPC comienza con la construcción de la correspondiente matriz de chequeo de paridad  $\mathbf{H}$ , a partir de la cual se obtiene una matriz de chequeo de paridad sistemática equivalente siguiendo la formulación de la matriz generadora  $\mathbf{G}$  del código. A partir de  $\mathbf{H}$ , se puede definir el síndrome de un vector de tamaño  $n \times 1$  como:

$$\mathbf{S} = \mathbf{H} \circ \mathbf{c} \quad (3.8)$$

lo que significa que cada vector código tiene que satisfacer la condición

$$\mathbf{H} \circ \mathbf{c} = \mathbf{0} \quad (3.9)$$

Como la matriz de chequeo de paridad  $\mathbf{H}$  es de dimensión  $(n - k) \times n$ , el síndrome corresponde a un vector de longitud  $n - k \times 1$ . De tal modo que cada elemento  $j$  del síndrome valdrá cero si y solo si el vector que multiplica a  $\mathbf{H}$  en la ecuación 3.8 cumple la ecuación de paridad representada por la fila  $j$  de la matriz  $\mathbf{H}$ . Esta forma alternativa de codificar y decodificar un código bloque indica que la matriz  $\mathbf{H}$  contiene toda la información necesaria para describir completamente el código bloque. Por otro lado, la expresión 3.7 será de ayuda para el diseño de un decodificador iterativo basado en el algoritmo suma-producto, que será descrito más adelante.

### 3.1.3 Descripción de los códigos LDPC

Los códigos LDPC más utilizados son normalmente códigos bloque lineales y binarios. Como se ha explicado anteriormente, el proceso de codificación se realiza multiplicando el vector de datos  $\mathbf{m}$  por la matriz generadora  $\mathbf{G}$  para obtener el vector de datos codificados  $\mathbf{c}$ . La matriz de chequeo de paridad  $\mathbf{H}$  es una matriz formada por vectores fila linealmente independientes que forman el subespacio dual del generado por los vectores fila de  $\mathbf{G}$ , los cuales también son linealmente independientes entre si. Por tanto, cada vector código va a satisfacer la condición  $\mathbf{H} \circ \mathbf{c} = \mathbf{0}$ .

El diseño de un código LDPC comienza por la construcción de la matriz de chequeo de paridad  $\mathbf{H}$ , que se caracteriza por ser *poco densa*. De acuerdo con la definición dada por Gallager, un código LDPC se denota como  $C_{LDPC}(n, s, v)$ , donde  $n$  es la longitud del código,  $s$  es el número de unos por columna, que generalmente será  $s \geq 3$ , y  $v$  es el número de unos por fila. Si las filas de la matriz de chequeo de paridad son linealmente independientes, entonces la tasa de código es igual a  $(v - s)/v$  [6]. Si no es así, la tasa de código será  $(n - s')/n$ , donde  $s'$  es la dimensión del subespacio vectorial generado por las filas de la matriz  $\mathbf{H}$ . Esta relación se obtiene contando el número total de unos por fila y por columna, obteniendo que  $ns = (n - k)v$ . A partir de esta expresión, se puede obtener fácilmente la tasa del código.

La construcción del código propuesto por Gallager le permitió demostrar las principales propiedades de estos códigos: la probabilidad de error decrece exponencialmente con el incremento de la longitud del código bloque, y la distancia

mínima del código también se incrementa conforme aumenta la longitud del código.

### 3.1.4 Construcción de los códigos LDPC

#### 3.1.4.1 Códigos regulares

El método de construcción propuesto por Gallager consistía en formar una matriz de chequeo de paridad  $\mathbf{H}$  poco densa determinando la posición de los unos aleatoriamente, con el número de unos por columna y por fila fijo; de manera que los códigos creados eran códigos LDPC regulares. Las condiciones que tienen que satisfacerse en la construcción de la matriz de chequeo de paridad  $\mathbf{H}$  de un código LDPC regular son [9]:

- La matriz de chequeo de paridad  $\mathbf{H}$  debe tener un número  $v$  fijo de unos por fila.
- La matriz de chequeo de paridad  $\mathbf{H}$  debe tener un número  $s$  fijo de unos por columna.
- El solapamiento de unos por columna y por fila debe ser como máximo igual a uno. Esta es una condición necesaria para evitar la presencia de ciclos en el correspondiente grafo bipartito.
- Los parámetros  $s$  y  $v$  deben ser números pequeños en comparación con la longitud del bloque.

Sin embargo, es muy difícil satisfacer la tercera condición si la intención es construir buenos códigos LDPC, porque los ciclos o bucles son inevitables en el grafo bipartito de un código LDPC eficiente [10].

Resumiendo el método de diseño de un código LDPC, primero se construye la matriz de chequeo de paridad poco densa  $\mathbf{H}$ , que tendrá la forma  $\mathbf{H} = [\mathbf{A}\mathbf{B}]$ , de acuerdo a las condiciones de construcción señaladas anteriormente. En general, esta matriz inicial no está en forma sistemática. La submatriz  $\mathbf{A}$  es una matriz cuadrada de dimensión  $(n - k) \times (n - k)$  que es no singular, esto es, que tiene matriz inversa  $\mathbf{A}^{-1}$ . La submatriz  $\mathbf{B}$  es de dimensión  $(n - k) \times k$ .

Mediante eliminación gaussiana, se puede transformar la matriz  $\mathbf{H} = [\mathbf{A}\mathbf{B}]$  en  $\mathbf{H}' = [\mathbf{I}_k \mathbf{A}^{-1}\mathbf{B}] = [\mathbf{I}_k \mathbf{P}^T]$ . Una vez que se ha formado la matriz de chequeo de paridad sistemática equivalente  $\mathbf{H}'$ , la correspondiente matriz generadora  $\mathbf{G}$  se puede construir usando las submatrices obtenidas, obteniéndose  $\mathbf{G} = [\mathbf{P}\mathbf{I}_k]$ . De esta forma, ambas matrices (la matriz generadora y la de chequeo de paridad) están definidas y el código LDPC está finalmente diseñado.

Los códigos LDPC pueden clasificarse, de acuerdo al método de construcción usado, en [11]:

- códigos LDPC aleatorios y
- códigos LDPC estructurados

En general, los códigos LDPC aleatorios obtienen resultados en términos de BER ligeramente superiores a los códigos estructurados, pero estos últimos códigos son menos complejos de codificar que los primeros.

### 3.1.4.2 Códigos irregulares

Un código LDPC irregular es aquél cuya matriz de chequeo de paridad  $\mathbf{H}$  tiene un número variable de unos por fila y por columna; esto es, no todas las filas o todas las columnas tienen el mismo número de unos. En general, los resultados en términos de BER para códigos LDPC irregulares son mejores que para códigos regulares. Hay varios métodos de construcción para códigos LDPC irregulares [9].

## 3.2 Codificación LDPC

La codificación LDPC va a consistir en, como ya se dijo anteriormente, la construcción de la matriz de chequeo de paridad  $\mathbf{H}$  así como la determinación de la secuencia de paridad  $p$  dada la secuencia de información  $s$ . A continuación van a exponerse tanto el proceso utilizado para generación de la matriz  $\mathbf{H}$  de paridad (la cual se obtendrá a partir de matrices modelo base especificadas para el estándar WiMAX y para cada ratio de código) como el proceso de generación del bloque codificado a partir de la obtención de los bits de paridad.

### 3.2.1 Generación de la matriz de paridad

El primer paso para la obtención de nuestro codificador LDPC consistirá en generar la matriz de chequeo de paridad  $\mathbf{H}$ , siguiendo el proceso estandarizado para WiMAX que se detalla a continuación.

Cada código LDPC puede ser definido por una matriz  $\mathbf{H}$  de tamaño  $m \times n$  donde  $n$  es la longitud del código y  $m$  es el número de bits de chequeo de paridad del código. La matriz  $\mathbf{H}$  es definida como:

$$\mathbf{H} = \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & \dots & P_{0,n_b-2} & P_{0,n_b-1} \\ P_{1,0} & P_{1,1} & P_{1,2} & \dots & P_{1,n_b-2} & P_{1,n_b-1} \\ P_{2,0} & P_{2,1} & P_{2,2} & \dots & P_{2,n_b-2} & P_{2,n_b-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{m_b-1,0} & P_{m_b-1,1} & P_{m_b-1,2} & \dots & P_{m_b-1,n_b-2} & P_{m_b-1,n_b-1} \end{bmatrix} = [\mathbf{P}^{H_b}] \quad (3.10)$$

donde  $P_{i,j}$  es una matriz de tamaño  $z \times z$  que o bien pertenece a un conjunto de matrices de permutación o es una matriz nula de dimensión  $z \times z$ . La matriz  $\mathbf{H}$  es una expansión de la matriz binaria base  $\mathbf{H}_b$  de dimensión  $m_b \times n_b$ , donde  $n = z \cdot n_b$  y  $m = z \cdot m_b$ , con  $z$  un entero al que llamaremos factor de expansión. El tamaño  $n_b$  de la

matriz base será un entero igual a 24.

Las permutaciones usadas (3.10) son desplazamientos circulares a la derecha, además el conjunto de matrices de permutación contienen la matriz identidad de dimensión  $z$   $z$  y versiones desplazadas circularmente a la derecha de la matriz identidad. Debido a que cada matriz de permutación es especificada por un desplazamiento circular a la derecha único, la información de la matriz base binaria y la información de permutación pueden ser combinada en una única matriz modelo  $H_{bm}$ .

La matriz base  $H_b$  binaria la obtendremos a partir de una serie de matrices modelo expandidas  $H_{bm}$  propias del estándar 802.16. Las matrices modelo base están estandarizadas para todos los ratios de código que podremos utilizar (y que pueden ser consultadas en la Figura 3.2 [12]), y éstas pueden ser utilizadas para cualquier longitud de código  $n$  realizando la expansión a través del factor de expansión  $z$ , que puede calcularse como muestra la ecuación 3.11 [12], obteniendo así  $H_{bm}$ , la cual puede ser directamente expandida a  $H$ .

$$z = \frac{n}{24} \quad (3.11)$$

El conjunto de desplazamientos  $p(i, j)$  de la matriz modelo base es usado para determinar el tamaño de desplazamiento para otras longitudes de código con el mismo ratio de código. De este modo (siendo  $f$  el índice que especifica las longitudes de código para un ratio de código dado y el factor de expansión  $z_0=96$  para código de longitud  $n = 2304$ ), el conjunto de desplazamientos para un código y longitud dada es:

- Para ratios de código 1/2, 3/4 A y B, 2/3 B y 5/6:

$$H_{bm} \equiv p(f, i, j) = \begin{cases} p(i, j), & p(i, j) \leq 0 \\ \left\lfloor \frac{p(i, j)z_f}{z_0} \right\rfloor, & p(i, j) > 0 \end{cases} \quad (3.12)$$

- Para ratios de código 2/3 A:

$$H_{bm} \equiv p(f, i, j) = \begin{cases} p(i, j), & p(i, j) \leq 0 \\ \text{mod}(p(i, j), z_f), & p(i, j) > 0 \end{cases} \quad (3.13)$$

Una vez obtenida  $H_{bm}$  que, como puede observarse en 3.2, estará formada por una expansión de esta con valores de desplazamiento positivos y -1;  $H_b$  se obtendrá reemplazando los valores negativos (-1) de  $H_{bm}$  por 0, y cada tamaño de desplazamiento circular  $p(i, j) \geq 0$  será reemplazado por 1.

$H_b$  puede expresarse como la concatenación de dos matrices, donde  $H_{b1}$  corresponde a la parte sistemática del código (bits sistemáticos) y  $H_{b2}$  corresponde a los bits de chequeo de paridad:  $H_b = [(H_{b1})_{m_b x k_b} | (H_{b2})_{m_b x m_b}]$ .  $H_{b2}$ , a su vez, puede dividirse en dos secciones como muestra la expresión 3.14, donde  $h_b$  es un vector con peso impar (número de 1's impar), y  $H'_{b2}$  tiene una estructura diagonal dual donde los elementos son 1 para  $i=j$  e  $i=j+1$ . La matriz base presenta unos valores  $h_b(0) = 1$ ,

Rate 1/2:

```
-1 94 73 -1 -1 -1 -1 -1 55 83 -1 -1 7 0 -1 -1 -1 -1 -1 -1 -1 -1
-1 27 -1 -1 -1 22 79 9 -1 -1 -1 12 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 24 22 81 -1 33 -1 -1 -1 0 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1
61 -1 47 -1 -1 -1 -1 -1 65 25 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1
-1 -1 39 -1 -1 -1 84 -1 -1 41 72 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 46 40 -1 82 -1 -1 -1 79 0 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1
-1 -1 95 53 -1 -1 -1 -1 -1 14 18 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1
-1 11 73 -1 -1 -1 2 -1 -1 47 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1
12 -1 -1 -1 83 24 -1 43 -1 -1 -1 51 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1
-1 -1 -1 -1 -1 94 -1 59 -1 -1 -1 70 72 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1
-1 -1 7 65 -1 -1 -1 -1 39 49 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0
43 -1 -1 -1 -1 66 -1 41 -1 -1 -1 26 7 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0
```

Rate 2/3 A code:

```
3 0 -1 -1 2 0 -1 3 7 -1 1 1 -1 -1 -1 -1 1 0 -1 -1 -1 -1 -1 -1
-1 -1 1 -1 36 -1 -1 34 10 -1 -1 18 2 -1 3 0 -1 0 0 -1 -1 -1 -1 -1 -1
-1 -1 12 2 -1 15 -1 40 -1 3 -1 15 -1 2 13 -1 -1 -1 0 0 -1 -1 -1 -1 -1
-1 -1 19 24 -1 3 0 -1 6 -1 17 -1 -1 -1 8 39 -1 -1 -1 0 0 -1 -1 -1 -1 -1
20 -1 6 -1 -1 10 29 -1 -1 28 -1 14 -1 38 -1 -1 0 -1 -1 -1 0 0 -1 -1 -1
-1 -1 10 -1 28 20 -1 -1 8 -1 36 -1 9 -1 21 45 -1 -1 -1 -1 -1 0 0 -1
35 25 -1 37 -1 21 -1 -1 5 -1 -1 0 -1 4 20 -1 -1 -1 -1 -1 -1 -1 0 0
-1 6 6 -1 -1 -1 4 -1 14 30 -1 3 36 -1 14 -1 1 -1 -1 -1 -1 -1 -1 -1 0
```

Rate 2/3 B code:

```
2 -1 19 -1 47 -1 48 -1 36 -1 82 -1 47 -1 15 -1 95 0 -1 -1 -1 -1 -1 -1
-1 69 -1 88 -1 33 -1 3 -1 16 -1 37 -1 40 -1 48 -1 0 0 -1 -1 -1 -1 -1 -1
10 -1 86 -1 62 -1 28 -1 85 -1 16 -1 34 -1 73 -1 -1 -1 0 0 -1 -1 -1 -1 -1
-1 28 -1 32 -1 81 -1 27 -1 88 -1 5 -1 56 -1 37 -1 -1 -1 0 0 -1 -1 -1 -1
23 -1 29 -1 15 -1 30 -1 66 -1 24 -1 50 -1 62 -1 -1 -1 -1 -1 0 0 -1 -1 -1
-1 30 -1 65 -1 54 -1 14 -1 0 -1 30 -1 74 -1 0 -1 -1 -1 -1 -1 0 0 -1
32 -1 0 -1 15 -1 56 -1 85 -1 5 -1 6 -1 52 -1 0 -1 -1 -1 -1 -1 0 0
-1 0 -1 47 -1 13 -1 61 -1 84 -1 55 -1 78 -1 41 95 -1 -1 -1 -1 -1 -1 0
```

Rate 3/4 A code:

```
6 38 3 93 -1 -1 -1 30 70 -1 86 -1 37 38 4 11 -1 46 48 0 -1 -1 -1 -1
62 94 19 84 -1 92 78 -1 15 -1 -1 92 -1 45 24 32 30 -1 -1 0 0 -1 -1 -1
71 -1 55 -1 12 66 45 79 -1 78 -1 -1 10 -1 22 55 70 82 -1 -1 0 0 -1 -1
38 61 -1 66 9 73 47 64 -1 39 61 43 -1 -1 -1 -1 95 32 0 -1 -1 0 0 -1
-1 -1 -1 -1 32 52 55 80 95 22 6 51 24 90 44 20 -1 -1 -1 -1 -1 0 0
-1 63 31 88 20 -1 -1 -1 6 40 56 16 71 53 -1 -1 27 26 48 -1 -1 -1 -1 0
```

Rate 3/4 B code:

```
-1 81 -1 28 -1 -1 14 25 17 -1 -1 85 29 52 78 95 22 92 0 0 -1 -1 -1 -1
42 -1 14 68 32 -1 -1 -1 -1 70 43 11 36 40 33 57 38 24 -1 0 0 -1 -1 -1
-1 -1 20 -1 -1 63 39 -1 70 67 -1 38 4 72 47 29 60 5 80 -1 0 0 -1 -1 -1
64 2 -1 -1 63 -1 -1 3 51 -1 81 15 94 9 85 36 14 19 -1 -1 0 0 -1 -1
-1 53 60 80 -1 26 75 -1 -1 -1 -1 86 77 1 3 72 60 25 -1 -1 -1 -1 0 0
77 -1 -1 -1 15 28 -1 35 -1 72 30 68 85 84 26 64 11 89 0 -1 -1 -1 -1 0
```

Rate 5/6 code:

```
1 25 55 -1 47 4 -1 91 84 8 86 52 82 33 5 0 36 20 4 77 80 0 -1 -1
-1 6 -1 36 40 47 12 79 47 -1 41 21 12 71 14 72 0 44 49 0 0 0 0 -1
51 81 83 4 67 -1 21 -1 31 24 91 61 81 9 86 78 60 88 67 15 -1 -1 0 0
50 -1 50 15 -1 36 13 10 11 20 53 90 29 92 57 30 84 92 11 66 80 -1 -1 0
```

Figura 3.2: Matriz modelo  $H_{mb}$  para cada ratio de código

$h_b(m_b - 1) = 1$  (que corresponden a tamaños de desplazamiento iguales), y un tercer valor  $h_b(j)$ ,  $0 < j < (m_b - 1)$  igual a 1 que corresponde a un tamaño de desplazamiento único y el cual nos permitirá obtener la primera secuencia de los bits de paridad para

esa fila específica. Como veremos a continuación, con este primer vector de paridad podremos calcular el resto de vectores para la formación de los bloques de información codificados.

$$H_{b2} = [h_{b1} | H'_{b2}] = \left[ \begin{array}{c|ccc} h_b(0) & 1 & & & \\ h_b(1) & 1 & 1 & 0 & \\ \cdot & & 1 & & \\ \cdot & & & 1 & \\ \cdot & 0 & 1 & 1 & \\ h_b(m_b - 1) & & & & 1 \end{array} \right] \quad (3.14)$$

### 3.2.2 Codificación LDPC

El codificador LDPC va a recibir el bloque de información  $s$  y usará la matriz modelo  $H_{bm}$  para determinar los bits de chequeo de paridad. A continuación, va a ser expuesto el algoritmo de codificación implementado en este proyecto.

Para la codificación, el bloque de información  $s$  va a ser dividido en  $k_b = n_b - m_b$  grupos de  $z$  bits, al que denominaremos  $u$ ,  $u = \begin{bmatrix} u(0) & u(1) & \dots & u(k_b - 1) \end{bmatrix}$ , donde cada elemento de  $u$  es un vector columna de la forma:  $u(i) = \begin{bmatrix} s_{iz} & s_{iz+1} & \dots & s_{(i+1)(z-1)} \end{bmatrix}^T$

Usando la matriz modelo  $H_{bm}$ , la secuencia de paridad  $p$  se determina también en grupos de  $z$ . Al grupo de secuencias  $p$  las denominaremos  $v$ ,  $v = \begin{bmatrix} v(0) & v(1) & \dots & v(m_b - 1) \end{bmatrix}$ , donde cada elemento de  $v$  es un vector columna de la forma:  $v(i) = \begin{bmatrix} p_{iz} & p_{iz+1} & \dots & p_{(i+1)(z-1)} \end{bmatrix}^T$

El proceso de codificación, va a constar de dos pasos, un primero de inicialización donde se determinará el valor del vector  $v(0)$  y un segundo de recursión en el que se determinarán el resto de vectores  $v(i + 1)$  a partir del vector  $v(i)$ , para  $0 \leq i \leq m_b - 2$ . Para la fase de inicialización, conseguiremos obtener el primer grupo de bits de paridad a partir de la posición de  $h_b$  en la que se hallaba el primer elemento no negativo y desaparejado en la columna correspondiente dentro de  $H_{bm}$  (lo que se traduce al uso de la posición en la que se encontraba el 1 intermedio de  $h_b$  cuando se hablaba de este vector como parte de la matriz  $H_b$ ). La expresión que será implementada para la obtención de  $v(0)$  se deriva de la suma de las filas de  $H_{bm}$  para obtener 3.15.

$$P_{p(x,k_b)} v(0) = \sum_{j=0}^{k_b-1} \sum_{i=0}^{m_b-1} P_{p(i,j)} u(j) \quad (3.15)$$

donde  $x$ ,  $1 \leq x \leq m_b - 2$ , como se ha comentado, es el índice de la fila de  $H_{bm}$  con el valor no negativo y único, y  $P_i$  representa la matriz identidad de dimensión  $z \times z$  circularmente desplazada a la derecha por el tamaño  $i$ .

La ecuación 3.16 se resuelve a través de la multiplicación de  $v_0$  por  $P_{p(x,k_b)}^{-1}$ , y

$P_{p(x,k_b)}^{-1} = P_{z-p(x,k_b)}$  donde  $p(x, k_b)$  representa el desplazamiento circular. Considerando la estructura de  $H'_{b2}$ , la recursividad se obtiene como

$$v(1) = \sum_{j=0}^{k_b-1} (P_{p(i,j)}u(j) + P_{p(i,k_b)})v(0), i = 0, \quad (3.16)$$

$$v(i+1) = v(i) + \sum_{j=0}^{k_b-1} P_{p(i,j)}u(j) + P_{p(i,k_b)}v(0), i = 1, \dots, m_b - 2 \quad (3.17)$$

donde  $P_{-1} \equiv 0_{zxz}$

Así, todos los bits de paridad que no se encuentran en  $v(0)$  se determinan evaluando la ecuación 3.17 para  $0 \leq i \leq m_b - 2$ .

### 3.3 Decodificación LDPC

#### 3.3.1 Decodificación de los códigos LDPC: El grafo de Tanner

Como se ha visto anteriormente, el vector de datos codificados se obtiene multiplicando el vector de datos  $\mathbf{m}$  por la matriz generadora matricial  $\mathbf{c} = \mathbf{G}^T \circ \mathbf{m}$ , donde

$$\mathbf{G}^T = \begin{bmatrix} \mathbf{P}^T \\ \mathbf{I}_k \end{bmatrix} \quad (3.18)$$

El vector transmitido se ve afectado por el ruido del canal, de modo que el vector recibido será  $\mathbf{r} = \mathbf{c} + \mathbf{n}$ , que es la entrada para los decodificadores tradicionales basados en el cálculo del síndrome  $\mathbf{S} = \mathbf{H} \circ \mathbf{r} = \mathbf{H} \circ \mathbf{G}^T \circ \mathbf{m} + \mathbf{n} = \mathbf{H} \circ \mathbf{n}$ . En esta sección se va a introducir un algoritmo de decodificación alternativo también basado en el cálculo del síndrome. La esencia de este algoritmo es determinar un vector  $\mathbf{d}$  que constituya una estimación del vector  $\mathbf{c}$  a partir del vector  $\mathbf{r}$  que satisfaga la condición  $\mathbf{H} \circ \mathbf{d} = \mathbf{0}$ . Este algoritmo es conocido como el *algoritmo suma-producto* y es en el que se basa este proyecto.

Este algoritmo determina la probabilidad a posteriori de cada símbolo del mensaje como una función de la señal recibida, la información del código, expresado en el caso de los códigos LDPC como las ecuaciones de paridad, y las características del canal. Este algoritmo se puede visualizar gráficamente mediante el grafo de Tanner [8], el cual representa la relación entre dos tipos de nodos, los nodos símbolo  $d_j$ , que representan los símbolos o bits transmitidos, y los nodos de chequeo de paridad  $h_j$ , que representan las ecuaciones de paridad. Las filas de la matriz de chequeo de paridad  $\mathbf{H}$  identifica los símbolos involucrados en cada ecuación de paridad, de modo que cada fila describe una ecuación de chequeo de paridad, y las posiciones marcadas con unos determinan las posiciones de los símbolos involucrados en dicha ecuación. De esta forma, y para códigos LDPC binarios, si la entrada  $i, j$  de la matriz de chequeo de paridad  $\mathbf{H}$  es igual a uno,



$H_{i,j}=1$ , entonces existe en el grafo de Tanner una conexión entre el nodo símbolo  $d_j$  y el nodo de chequeo de paridad  $h_i$ ; para cualquier otro caso, esta conexión no está presente.

El estado de un nodo de chequeo de paridad dado depende de los valores de los nodos símbolo conectados a él. En general, los nodos de chequeo de paridad conectados a un nodo símbolo dado se llaman *los nodos hijos* de ese nodo símbolo, y los nodos símbolo conectados a un nodo de chequeo de paridad dado se llaman *los nodos padres* de ese nodo de chequeo de paridad.

En el algoritmo suma-producto, cada nodo símbolo  $d_j$  envía a cada uno de sus nodos de chequeo de paridad hijo  $h_i$  una estimación  $Q_{ij}^x$  de la probabilidad de que el nodo de chequeo de paridad esté en el estado  $x$  (valga 0 o 1), basándose en la información dada por los otros nodos hijo de ese nodo símbolo. Por otro lado, cada nodo de chequeo de paridad  $h_i$  envía a cada uno de sus nodos símbolo padre  $d_j$  una estimación  $R_{ij}^x$  de la probabilidad de que la ecuación de paridad  $i$  relativa al nodo  $h_i$  sea satisfecha si el nodo símbolo o padre está en el estado  $x$ , teniendo en cuenta la información dada por todos los demás nodos padre conectados a ese nodo de chequeo de paridad. Esto puede verse en el gráfico de Tanner representado en la Figura 3.3.

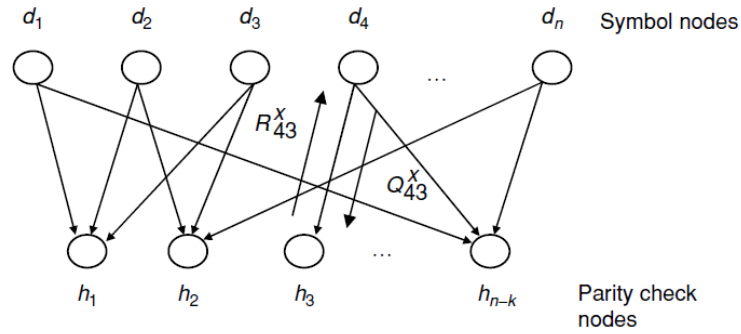


Figura 3.3: Grafo de Tanner. Grafo bipartito que une los nodos símbolo con los nodos de chequeo de paridad

Este es un proceso iterativo de intercambio de información entre los dos tipos de nodos en el grafo bipartito. El proceso iterativo se para si, después del cálculo de la condición del síndrome sobre el vector decodificado  $\mathbf{d}$ , en una iteración dada, el resultado del vector síndrome es el vector de todo ceros. Si después de varias iteraciones sucesivas el síndrome no llega a ser el vector de todo ceros, el decodificador se para cuando se alcanza un determinado número de iteraciones preestablecido con antelación. En ambos casos, el decodificador genera de manera óptima bits o símbolos decodificados, pero éstos no formarán un vector código si el síndrome obtenido no ha sido cero. En este sentido el algoritmo suma-producto actúa de la misma forma que el algoritmo MAP (Maximum A Posteriori), definiendo la mejor estimación posible para cada símbolo del vector recibido, pero no necesariamente definiendo la mejor estimación del vector código entero que inicialmente fue transmitido a través del canal.

En general, la decodificación iterativa de los códigos LDPC converge a la información verdadera del mensaje cuando el correspondiente grafo bipartito tiene una estructura de

árbol, esto es, no contiene ciclos o bucles. El efecto de degradación de los ciclos de longitud corta en el gráfico bipartito disminuye cuando la longitud del código aumenta y se reduce considerablemente si la longitud del código es muy grande ( $\geq 1000$  bits).

### 3.3.2 Algoritmo suma-producto simplificado

Para la decodificación LDPC vamos a usar el *algoritmo suma-producto simplificado*. Como su propio nombre indica, se trata de una simplificación del algoritmo iterativo suma-producto que fue introducido anteriormente.

El objetivo del algoritmo de decodificación suma-producto simplificado es encontrar el vector decodificado  $\mathbf{d}$  como una estimación del vector código transmitido  $\mathbf{c}$  y que satisfaga la condición del síndrome.

En el algoritmo suma-producto, cada nodo símbolo  $d_j$  manda a cada nodo hijo de chequeo de paridad  $h_i$  la estimación  $Q_{ij}^x$  (basada en la información dada por el resto de sus nodos hijo de chequeo de paridad) de que el correspondiente nodo de chequeo de paridad se encuentra en el estado  $x$ . Por otro lado, cada nodo de chequeo de paridad  $h_i$  manda a cada nodo símbolo padre  $d_j$  la estimación  $R_{ij}^x$ , que se calcula con la información dada por los otros nodos símbolo, de que la correspondiente ecuación de chequeo de paridad  $i$  sea satisfecha si el nodo símbolo está en el estado  $x$ .

En el algoritmo suma-producto, los valores de los coeficientes  $R_{ij}^0$  y  $R_{ij}^1$  son determinados como una función de los valores de los coeficientes  $Q_{ij}^0$  y  $Q_{ij}^1$  teniendo en cuenta todas las combinaciones de los bits código que satisfacen la ecuación de chequeo de paridad relacionada con ese cálculo. Sin embargo, MacKay y Neal [6] introdujeron un método de cálculo que permite evitar tener en cuenta todas estas posibilidades de la ecuación de chequeo de paridad en el cálculo de los valores de los coeficientes  $R_{ij}^0$  y  $R_{ij}^1$ : el *algoritmo suma-producto simplificado*.

Este algoritmo requiere un proceso de inicialización que consiste en determinar los valores  $Q_{ij}^x$ . Estos valores se escogen con la estimación *a priori* de los símbolos recibidos, denotados como  $f_j^x$  (probabilidad de que el símbolo  $j$  sea  $x$ ):

$$Q_{ij}^0 = f_j^0 \text{ y } Q_{ij}^1 = f_j^1 \quad (3.19)$$

con

$$f_j^1 = \frac{1}{1 + e^{-\frac{2Ay_j}{\sigma^2}}} \quad (3.20)$$

y

$$f_j^0 = 1 - f_j^1 \quad (3.21)$$

donde  $y_j$  es la salida del canal en el instante  $j$ , delta cuadrado es la potencia de ruido

y los bits son transmitidos en formato polar con amplitudes  $\pm A$ .

Esta versión simplificada, lleva a cabo el cálculo iterativo implementando dos pasos, el paso horizontal y el paso vertical, de acuerdo a la forma en que los valores son tomados de la correspondiente matriz de chequeo de paridad  $\mathbf{H}$  (por filas o por columnas).  $\delta Q_{ij}$  se define como:

$$\delta Q_{ij} = Q_{ij}^0 - Q_{ij}^1 \quad (3.22)$$

a su vez, se definen los términos  $\delta R_{ij}$ , los cuales se calculan del siguiente modo:

$$\delta R_{ij} = \prod_{j' \in N(i) \setminus j} \delta Q_{ij'} \quad (3.23)$$

donde  $N(i)$  representa el conjunto de índices de todos los nodos símbolo padre conectados al nodo de chequeo de paridad  $h_i$ , mientras  $N(i) \setminus j$  representa el mismo conjunto excluyendo el nodo símbolo padre  $d_j$ .

Los coeficientes  $R_{ij}^0$  y  $R_{ij}^1$  se calculan a partir de  $\delta R_{ij}$ :

$$R_{ij}^0 = 1/2(1 + \delta R_{ij}) \quad (3.24)$$

y

$$R_{ij}^1 = 1/2(1 - \delta R_{ij}) \quad (3.25)$$

Los coeficientes  $Q_{ij}^0$  y  $Q_{ij}^1$  se actualizan en el paso vertical. Su valor para cada posible valor de  $x$  (que en el caso binario puede ser 0 o 1) es:

$$Q_{ij}^x = \alpha_{ij} f_j^x \prod_{i' \in M(j) \setminus i} R_{i'j}^x \quad (3.26)$$

donde  $M(j)$  representa el conjunto de índices de todos los nodos hijo de chequeo de paridad conectados al nodo símbolo  $d_j$ , mientras  $M(j) \setminus i$  representa el mismo conjunto excluyendo el nodo hijo de chequeo de paridad  $h_i$ . El coeficiente  $f_j^x$  es la *probabilidad a priori* de que el nodo símbolo  $d_j$  esté en el estado  $x$ .  $\alpha_{ij}$  se selecciona de modo que se cumpla que  $Q_{ij}^0 + Q_{ij}^1 = 1$ .

Para la obtención de la estimación del vector decodificado en cada iteración, se requiere el cálculo de las *probabilidades a posteriori*  $Q_j^0$  y  $Q_j^1$ , que son iguales a:

$$Q_j^x = \alpha_j f_j^x \prod_{i \in M(j)} R_{ij}^x \quad (3.27)$$

donde, una vez más, la constante  $\alpha_j$  se elige de tal forma que se cumpla:  $Q_j^0 + Q_j^1 = 1$ . La estimación del vector decodificado  $\hat{\mathbf{d}}$  en cada iteración se puede obtener con la

expresión:

$$\hat{d}_j = \max(Q_j^x) \quad (3.28)$$

lo que equivale a:

$$\text{if } Q_j^0 > Q_j^1 \text{ then } \hat{d}_j = 0, \text{ else } \hat{d}_j = 1 \quad (3.29)$$

## Capítulo 4

# Arquitectura del sistema WiMAX

A continuación se explica la arquitectura del sistema WiMAX utilizado para la realización de este proyecto.

### 4.1 Descripción general del escenario utilizado

El escenario utilizado para la implementación de este proyecto se basa, principalmente, en un sistema digital de comunicaciones básico. Se trata de un entorno virtual, desarrollado en C++, en el que poder simular las prestaciones de un sistema WiMAX real con la elección de diferentes parámetros. El objeto de este proyecto ha sido la implementación e integración de un codificador-decodificador LDPC en este sistema así como la simulación y evaluación de los resultados obtenidos.

Como muestra la figura 4.1, el flujo de bits a la entrada del sistema va a ser generado de manera aleatoria y la longitud de dicho flujo de datos va a ser un parámetro a elegir. Las salidas más importantes de nuestro sistema serán el BER (probabilidad de error en el bit) obtenido para cada SNR, así como el BLER (probabilidad de error en el símbolo).

Como aparece en la figura, en el emisor se halla la fuente de información seguida del codificador de fuente, cuya finalidad es obtener una representación eficiente de los símbolos del alfabeto fuente de forma que los mensajes estén constituidos con el menor número de bits posible (o si se tratase de una fuente analógica también proporcionaría la conversión A/D). Sin embargo, para el estudio de la codificación y decodificación LDPC que permite WiMAX vamos a obviar el codificador de fuente y supondremos que la fuente ya proporciona al codificador de canal la secuencia de bits que representan eficientemente la información que se desea transmitir.

A continuación, vamos a explicar la función de cada uno de los bloques de los que consta nuestro sistema (mostrados en 4.1).

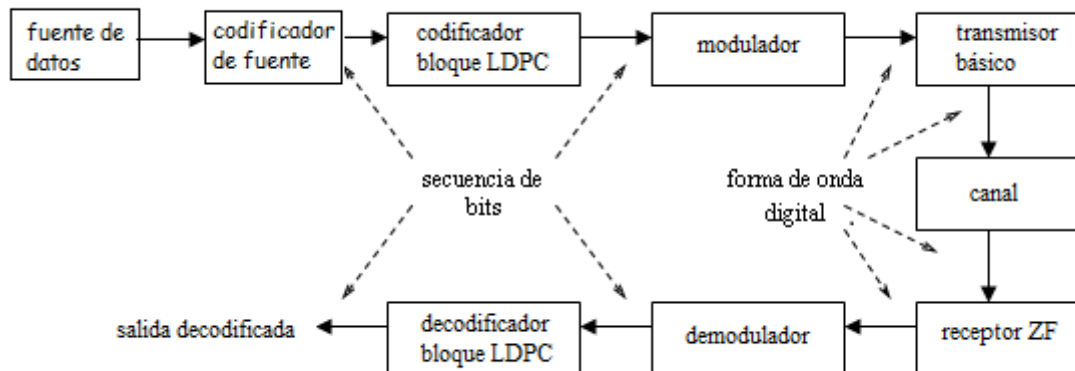


Figura 4.1: Diagrama del sistema de comunicaciones WiMAX utilizado

#### 4.1.1 Modulador

Vamos a describir brevemente el bloque correspondiente al modulador. Lo que pretende la modulación digital es modular una señal analógica con una secuencia digital para poder transportar la información a través de un medio, que en nuestro caso se trata de un canal radio móvil. El estándar WiMAX/802.16 nos permite modular la información con estos 3 tipos de modulaciones digitales: QPSK, 16QAM y 64QAM. Tener más de una posible modulación tiene la gran ventaja de que puede aplicarse la estrategia de modulación adaptativa, proceso que ya es usado en otros sistemas de comunicaciones como GSM, UMTS, WiFi, etc. La modulación adaptativa puede combinarse también con una tasa de codificación adaptativa, dando lugar a lo que se conoce como la técnica ACM (*Adaptive Coding and Modulation*), que es capaz de incrementar la capacidad global del sistema y maximizar la tasa de transferencia para la SNR disponible, permitiendo una solución de compromiso en tiempo real entre tasa de transferencia y robustez. La técnica ACM se realiza entre el dispositivo móvil y la estación base por medio de un canal “feedback” que indica la calidad del canal.

El principio de funcionamiento de ACM es simple: cuando el enlace radio es bueno (alta SNR y no hay desvanecimientos) usa una modulación de alto nivel para aprovechar la situación y aumentar así la eficiencia espectral, además de usar una tasa de codificación alta, que aumenta la tasa de datos de información, debido a que no son necesarios tantos bits redundantes para una decodificación correcta; mientras que cuando el canal radio presenta malas condiciones (baja SNR y existen desvanecimientos de señal) usa una modulación de bajo nivel pero muy robusta, ya que los símbolos en su constelación están más alejados entre sí y será más difícil que se obtengan errores en el receptor a cambio de perder eficiencia espectral. En el caso de encontrarse en un mal enlace radio, la tasa de codificación elegida será baja para así aumentar el número de bits de paridad y tener más capacidad para poder corregir los errores introducidos por el canal.

### 4.1.2 Transmisor básico

Puesto que nuestro sistema se trata de un sistema de comunicaciones WiMAX que utiliza una modulación OFDM para transmitir la información a través del canal, no nos basta con modular la secuencia de entrada con modulación QPSK, 16QAM o 64QAM; si no que es necesario la conformación de símbolos OFDM con estos datos modulados para su envío a través del canal.

Los símbolos una vez modulados son mapeados en las subportadoras OFDM, siendo éste el último paso antes de la transmisión propiamente dicha. OFDM es una técnica de transmisión muy potente que está basada en la transmisión simultánea de muchas frecuencias ortogonales de banda estrecha, lo que elimina la interferencia entre canales, denominadas subportadoras. El número de subportadoras es normalmente denotado por  $N$ . El ancho de banda frecuencial asociado a cada uno de los canales es entonces mucho menor que si el ancho de banda total fuese ocupado por una sola modulación (Single Carrier). El hecho de que las portadoras sean ortogonales y que posea una buena resistencia a la propagación multicamino permite a OFDM tener una alta eficiencia espectral y ser considerada la mejor técnica de transmisión para sistemas wireless en los que precisamente la propagación multicamino está muy presente. Algunas de las características de los sistemas OFDM son:

- La tasa de transmisión de datos se divide en tasas de transmisión menores para cada una de las subportadoras que es modulada a cada una de las frecuencias ortogonales.
- Debido a esta división frecuencial, el ancho de banda ocupado por cada subportadora será mucho menor en comparación al ancho de banda total. Por ello, esto hace que un canal con desvanecimiento selectivo en frecuencia pase a ser un canal con desvanecimiento plano, lo que evita la aparición de ISI (*Interferencia Intersimbólica*) y la más fácil ecualización de ésta.
- OFDM introduce también un prefijo cíclico CP en las muestras en el dominio temporal para combatir el efecto del multicamino.
- Todo el sistema se puede realizar con IFFT en el transmisor, y con FFT en el receptor donde cada subportadora puede ser tratada con independencia, eliminando la complejidad del ecualizador.

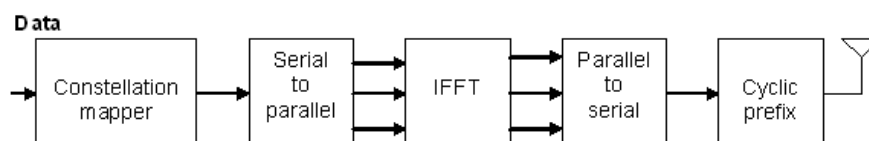


Figura 4.2: Generación de un símbolo OFDM

El principio de funcionamiento de OFDM consiste en aplicar la IFFT (*Inverse Fast Fourier Transform*) a  $N$  subportadoras para generar de esta forma una señal OFDM, o dicho de otra manera, un símbolo OFDM de  $N$  muestras temporales y duración  $T_d$ . Estos  $N$  símbolos tomados como coeficientes frecuenciales por la IFFT son los correspondientes a los datos modulados en QPSK, 16QAM o 64QAM. Una simplificación de este proceso de generación de símbolos OFDM se presenta en la Figura 4.2.

Después de la aplicación de la IFFT se añade un Prefijo Cíclico (Cyclic Prefix, CP) al principio del símbolo OFDM, cuya duración es llamada Tiempo de Guarda (TG). El tiempo total llamado Tiempo de Símbolo, por tanto, es  $T_S = T_G + T_d$ . El CP permite al receptor absorber mucho más eficientemente el *delay spread* debido al multitrayecto y mantener la ortogonalidad frecuencial. La relación entre TG y  $T_d$  es denotada como  $G = T_G / T_d$ , cuyos posibles valores definidos por el estándar 802.16 son:  $1/32$ ,  $1/16$ ,  $1/8$  y  $1/4$ . Hay que tener en cuenta que si es necesario emplear un valor alto de  $G$  debido a que el efecto multicamino es importante, se estará disminuyendo la tasa de datos útiles. La elección del valor de  $G$  adecuado para cada situación se toma por la estación base.

#### 4.1.3 Canal: Tipo y parámetros

Este bloque va a ser el encargado de simular las condiciones del canal por el que se transmite la información. Dicho bloque nos va a permitir general diversos tipos de canales entre los que se encuentran los que van a ser utilizados para el estudio en este proyecto.

El primero de ellos es el canal gaussiano AWGN, canal ideal, compuesto por un sólo rayo de potencia unidad y sin ningún tipo de retardo. El segundo de los canales que utilizaremos, será el canal ITU vehicular A extendido que, como su nombre indica, nos va a simular las condiciones que se tendrían al recibir la información a bordo de un medio de transporte con una cierta velocidad. Este segundo canal, estará formado por varios rayos (fruto de los diferentes caminos que va a seguir la señal) de una determinada potencia cada uno de ellos. Por último, el tercero de los canales a utilizar es el ITU Pedestrian A extendido que nos simulará las condiciones del medio que afectarán a nuestra señal cuando la información va a ser recibida por un usuario que se mueve. En la tabla 4.1 se muestran los valores de los parámetros más representativos para cada canal.

Tipo de canal	Número de rayos	Potencia media cada rayo en dBs	Retardo de cada rayo (ns)
ITU vehicular A extendido	9	0.0 -1.5 -1.4 -3.6 -0.6 -9.1 -7.0 -12.0 -16.9	0 30 150 310 370 710 1090 1730 2510
ITU pedestrian A extendido	7	0.0 -1.0 -2.0 -3.0 -8.0 -17.2 -20.8	0 30 70 90 110 190 410

Tabla 4.1: Valores seleccionados para cada canal

Básicamente, el efecto que el canal produce en la información transmitida es un



cambio en la amplitud de los símbolos transmitidos así como un cambio en la fase (debido a las diferencias de caminos (rayos) que se producen) que va a provocar el fenómeno conocido como ISI (Interferencia Intersimbólica). En resumen, un canal wireless queda caracterizado por:

- Diversas pérdidas en el trayecto del rayo (producidas por las características del terreno: urbano o rural, vegetación, etc; medio de propagación: lluvia, la distancia entre el transmisor y el receptor, etc)
- Efecto multitrayecto, que se traduce en retrasos en la señal de información recibida.
- Fading multitrayecto: debido al efecto multitrayecto, la señal recibida experimenta fluctuaciones en su amplitud, fase y ángulo de llegada.
- Interferencia intersimbólica (**ISI**) producida por los diferentes retardos de cada rayo
- Efecto Doppler en la señal transmitida (cuando el receptor de la señal está en movimiento)

#### 4.1.4 Receptor ZF

Tras el paso por el canal, la información ha de ser recuperada por el bloque **receptor\_ZF**. Este bloque va a encargarse de recibir los datos OFDM modulados y degradados (ruido en la señal recibida así como desvanecimientos en la señal) por el efecto del canal y de intentar recuperar la señal modulada enviando ecualizando en la medida de lo posible el efecto del canal. La ecualización consiste en compensar la respuesta frecuencial del canal por el que ha sido transmitida la señal de datos mediante un filtro diseñado para ello. Para el diseño de este filtro, existe un sistema adaptativo de estimación del canal que define los coeficientes del filtro; de esta manera, el ecualizador se adapta automáticamente a las propiedades del canal de comunicación variantes en el tiempo tales como la propagación multicamino, los desvanecimientos o el ruido, y mitigando los efectos de la ISI.

Posteriormente a la ecualización, se van a recuperar los símbolos modulados QPSK, 16QAM o 64QAM realizando la función inversa que se realizó para la conformación de los símbolos OFDM: FFT, de modo que ya puedan ser demodulados.

#### 4.1.5 Demodulador

Una vez recibida la señal y ecualizada, hay que proceder a la demodulación de la información recibida. Como parámetros de entrada a este bloque tendremos los mismos que se especificaron para el modulador, de modo que la demodulación escogida será en base a estos parámetros. Una vez escogida el tipo de modulación (QPSK, 16QAM o 64QAM), el sistema da la opción de realizar dos tipos de demodulación:

- Demodulación hard: La decisión de los símbolos recibidos depende de unas constantes propias para cada demodulación.
- Demodulación soft: La decisión de los símbolos recibidos se realizará de acuerdo a las distancias entre cada símbolo recibido y cada uno de los símbolos perteneciente a la constelación de esa modulación.

Debido a la gran información que la demodulación soft aporta y, puesto que necesitamos unas probabilidades a priori para nuestro decodificador, ésta va a ser la demodulación escogida para la implementación y evaluación de este proyecto.

## Capítulo 5

# Propuesta de codificador/decodificador bloque para sistemas WiMAX: el código LDPC

El enlace radio es un tipo de canal que varía muy rápidamente y que a menudo sufre grandes interferencias. La codificación de canal, cuyas funciones principales son prevenir y corregir los errores de transmisión de los sistemas wireless, debe tener un rendimiento muy bueno con el fin de mantener altas velocidades de transmisión de datos. La cadena de codificación de canal de WiMAX se encuentra en la capa física del estándar 802.16 y está compuesta por tres pasos: randomización, codificador FEC y entrelazado (Figura 5.1) [13] [14]; aplicados sobre cada PHY PDU (PHYsical Protocol Data Units) en ese mismo orden en el transmisor. Las correspondientes operaciones son realizadas en el receptor pero en el orden inverso. A los datos que se codifican en el emisor, previamente se les ha añadido los bits propios de la detección de errores a través de un código CRC. Estos bits del código CRC son considerados también como bits de información a transmitir.

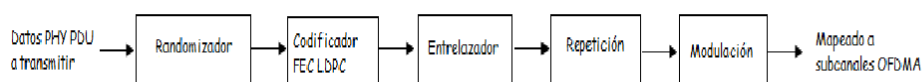


Figura 5.1: Diagrama de bloque del proceso de codificación estipulado por el estándar WiMAX

Uno de los codificadores FEC especificado en este estándar es un turbocódigo convolucional (CTC) constituido por un código duo-binario convolucional, sistemático, recursivo y circular. El principal inconveniente de este tipo de codificadores es su elevada complejidad ya que se trata de una doble codificación (codificación FEC y entrelazado).

Lo que este proyecto pretende es analizar y encontrar una codificación alternativa para este tipo de sistemas, analizando sus prestaciones en términos de capacidad correctora de errores y carga computacional disminuyendo la complejidad en la implementación.

Por todo ello, vamos a implementar otro esquema de codificación bloque alternativo a los turbocódigos y que también es admitido por el estándar WiMAX 802.16e [14]: los códigos LDPC (Figura 5.2).

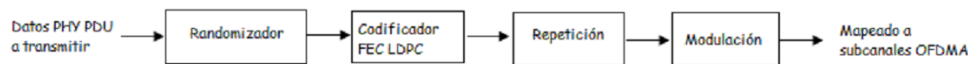


Figura 5.2: Diagrama de bloque del proceso de codificación propuesto para sistemas WiMAX

## 5.1 El codificador bloque LDPC

### 5.1.1 Randomizador

Como se muestra en la Figura 5.2, el primer elemento que va a tener nuestro codificador es un randomizador. Este bloque va a servir para introducir protección a los datos a través de la reordenación aleatoria de los bits de entrada (únicamente de los bits de información, excluyendo preámbulos y cabeceras). Su objetivo es el de maximizar la entropía de la fuente, esto es, igualar la probabilidad de transmisión de unos y ceros. Como consecuencia del proceso, las largas cadenas de unos y ceros consecutivos que pudieran existir en la secuencia binaria a transmitir se eliminan.

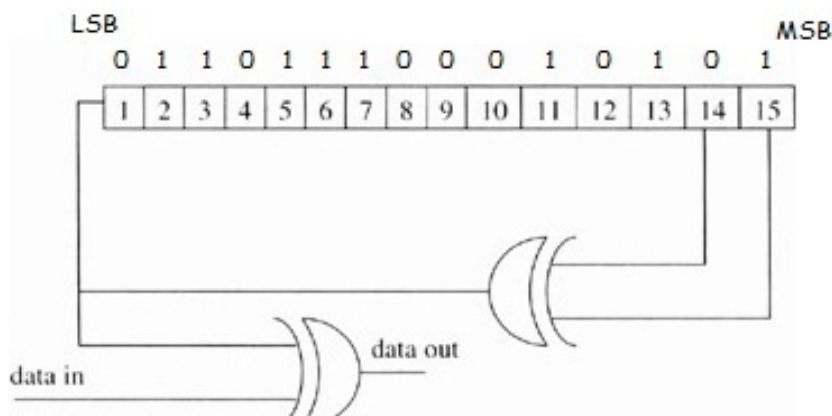


Figura 5.3: Generador PRBS para el proceso de aleatorización en WiMAX

En nuestro proceso de aleatorización vamos a utilizar un generador pseudoaleatorio o PRBS (*Pseudo Random Binary Sequence*) y que va a ser aplicado a cada ráfaga de

datos a transmitir tanto en el downlink como en el uplink. El randomizador PRBS (Figura 5.3) está formado por un registro de desplazamiento de 15 bits que se inicializa con una secuencia formada por el identificador de la estación base (BSID), el DIUC (Downlink Interval Usage Code) o el UIUC (Uplink Interval Usage Code) (dependiendo de si estamos en el enlace ascendente o descendente) y el número de trama que se esté transmitiendo (en la Figura 5.4 puede verse el proceso de formación de las secuencias de inicialización para ambos canales). En nuestro caso siempre vamos a inicializar con la misma secuencia: [LSB]011011100010101[MSB]

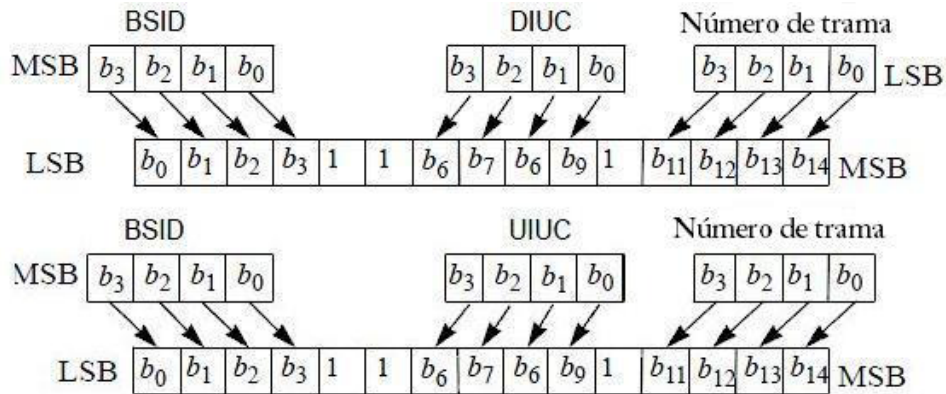


Figura 5.4: Proceso de formación de las secuencias de aleatorización para el randomizador en el enlace descendente (arriba) y el ascendente (abajo)

### 5.1.2 El codificador

Como ya se expuso en el capítulo 3, la obtención de la matriz de chequeo de paridad  $H$  binaria se conseguirá a partir de la implementación en C++ del proceso descrito en la sección 3.2.1 (en el cual se establecen una serie de matrices modelo base propias del estándar 802.16) atendiendo también a la longitud del código.

Por otro lado, la implementación de las expresiones expuestas en la sección 3.2.2 permitirán la obtención del vector codificado, el cual estará formado por los bits de salida del bloque aleatorizador seguidos de los bits de paridad (fruto de la implementación de las expresiones 3.15 y 3.17). Ésta será la secuencia resultante que será enviada a través del canal de comunicación WiMAX.

## 5.2 El decodificador bloque LDPC

El primer paso para la decodificación LDPC de los bloques recibidos consiste en dealeatorizar los datos recibidos, puesto que estos fueron aleatorizados para introducir protección en el proceso de codificación. Para ello utilizaremos el mismo randomizador que se utilizó previo a la codificación y que fue explicado en el apartado 5.1.1 de este mismo capítulo.

El decodificador LDPC que ha sido implementado en C++ para el sistema WiMAX

de capa física es el *algoritmo iterativo suma-producto simplificado* que ya fue explicado en el apartado 3.3 del capítulo 3.

Como ya se dijo, este algoritmo necesita de unas *probabilidades a priori* para poder realizar la inicialización de sus coeficientes con los que llevar a cabo dicha decodificación. Las *probabilidades a priori* que necesitamos conseguir, son las correspondientes a la expresión 5.1. Para ello se va a hacer uso de los valores soft suministrados por el demodulador, estos valores soft corresponden a las LLR (Log Likelihood Ratio) de los bits demodulados, por lo que se pueden usar para obtener las probabilidades a priori que el decodificador necesita. Las LLR se definen como muestra la expresión 5.2

$$f_j^1 = P(r_j|y_j = 1) \text{ y } f_j^0 = P(r_j|y_j = 0) \quad (5.1)$$

$$LLR = L(r_j|y_j) = \ln \frac{P(r_j|y_j = 1)}{P(r_j|y_j = 0)} \quad (5.2)$$

Donde  $r_j$  es el símbolo soft demodulado en el instante de tiempo  $j$  e  $y_j$  es el bit  $j$  de información original transmitido. Como se ve en la expresión 5.2, el LLR correspondiente al logaritmo del cociente entre la probabilidad condicionada de que se reciba  $r_j$  habiendo transmitido el bit  $y_j$  igual a 1, y la probabilidad condicionada de que se reciba  $r_j$  si el correspondiente bit  $y_j$  transmitido ha sido un 0.

A la vista de esto, si existe una mayor probabilidad de que  $r_j$  corresponda a un bit transmitido  $y_j = 1$ , el ratio LLR tomará un valor positivo; mientras que si es mayor la probabilidad de que corresponda a  $y_j = 0$ , el ratio LLR será negativo. Si se tuviese la certeza de que se trata de un 1, entonces el ratio LLR valdría  $+\infty$ ; mientras que si se tuviese la certeza de que el bit transmitido fue un 0, el ratio LLR tomaría el valor  $-\infty$ .

De modo que las probabilidades a priori con las que vamos a inicializar nuestro decodificador serán (expresiones 5.3 y 5.4):

$$P(r_j|y_j = 1) = e^{LLR} P(r_j|y_j = 0) \text{ y } P(r_j|y_j = 0) = 1 - P(r_j|y_j = 1) \quad (5.3)$$

de modo que:

$$P(r_j|y_j = 1) = \frac{e^{LLR}}{1 + e^{LLR}} \text{ y } P(r_j|y_j = 0) = 1 - \frac{e^{LLR}}{1 + e^{LLR}} \quad (5.4)$$

Una vez que el algoritmo comienza con la decodificación, se van a realizar todas las operaciones necesarias (las cuales fueron introducidas en el apartado 3.3) para la ejecución de los pasos horizontal y vertical en cada iteración. Asimismo, el *algoritmo suma-producto simplificado* calculará una estimación del vector decodificado para cada iteración. Este vector, además de contener los bits estimados que fueron enviados por el transmisor, será utilizado para calcular el síndrome y saber si ya hemos obtenido el vector decodificado correctamente. Puesto que habrá situaciones en las que nunca se llegue a obtener el vector decodificado sin ningún error, va a ser necesario definir un número máximo de iteraciones para el decodificador, de tal forma que, si se llega a este

número máximo aun cuando no se haya obtenido el vector decodificado correctamente, termine la codificación y podamos obtener la probabilidad de error en el bloque.





## Capítulo 6

# Simulación y análisis de resultados

### 6.1 Consideraciones generales de las simulaciones

En este capítulo van a ser presentados los resultados más significativos de todas las simulaciones realizadas a lo largo de este estudio. El entorno de simulación de capa física WiMAX utilizado (basado en C++) incluye todos los bloques descritos en los capítulos anteriores. Para la correcta comprensión y valoración de los datos que aquí van a ser presentados, hay que tener en cuenta una serie de aspectos tales como:

- Para la caracterización del canal wireless se han utilizado 3 modelos diferentes: AWGN (canal ideal gaussiano), y los modelos extendidos ITU Pedestrian A y Vehicular A. Estos dos últimos modelos corresponden a un canal multicamino con desvanecimientos adaptado a una transmisión OFDM a velocidades propias de un peatón y un vehículo. Dichas velocidades se han establecido en 3 Km/h y 120 Km/h respectivamente para realizar la simulación del comportamiento del sistema considerando las condiciones de transmisión que afectan a los usuarios del sistema WiMAX en una situación normal.
- El ecualizador previo a la demodulación es un *Zero Forcing*, consistente en la inversión de la respuesta frecuencial del canal. Se ha optado por este tipo de ecualización debido a su sencillez en la implementación ya que se ha supuesto estimación ideal del canal.
- Se debe considerar que un canal wireless varía constantemente a lo largo del tiempo; lo que supone que para cada bloque transmitido las condiciones de transmisión son diferentes, es decir, el canal es independiente y no constante entre transmisiones. Sin embargo, en el caso del modelo de canal ITU Pedestrian A puede aproximarse a canal constante ya que, puesto que las transmisiones se envían en la misma zona del espectro de la señal OFDM, a la velocidad de 3 Km/h el tiempo de coherencia del canal es relativamente elevado con respecto al tiempo entre transmisiones. Este

hecho va a provocar que disminuya la diversidad del canal, por lo que se pueden obtener resultados peores que para canales vehiculares.

Con el fin de analizar en profundidad el comportamiento que los códigos LDPC tienen en este tipo de sistemas wireless, van a ser propuestos tres estudios (que van a ser expuestos a continuación) en los que se presentarán los resultados de las simulaciones en gráficas que muestran la tasa de error en el bloque (BLER) en función de la SNR.

## 6.2 Efecto del tamaño de bloque FEC en la decodificación LDPC

El primero de los estudios tiene por objeto la evaluación del efecto que la elección del tamaño de bloque FEC tiene en la decodificación en términos del BLER obtenido en función de la SNR. La tabla 6.1 muestra los parámetros de simulación que van a ser utilizados para este estudio.

Tipo de canal	AWGN, Pedestrian A y Vehicular A
Tipo de Modulación	QPSK
Tasa de código	1/2
Tamaño de bloque FEC	36,48,72,108 y 144 bytes
Número iteraciones del decodificador	50

Tabla 6.1: Parámetros de la simulación para el estudio del efecto del tamaño de bloque FEC

A continuación se muestran las gráficas con la BLER generada por el sistema con los parámetros seleccionados y los tres tipos de canal seleccionados: AWGN (Figura 6.1), Pedestrian A (Figura 6.2) y Vehicular A (Figura 6.3).

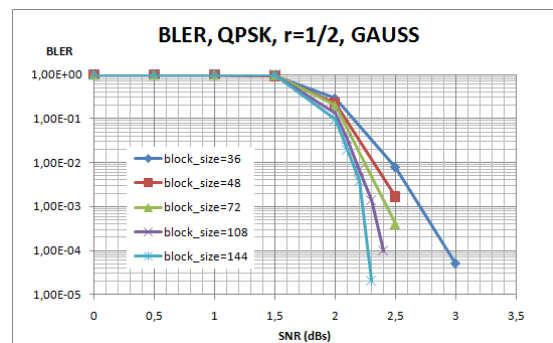


Figura 6.1: BLER para QPSK, tasa 1/2 y canal AWGN

A la vista de estos resultados, pueden sacarse las siguientes conclusiones importantes:

- Como cabía esperar, se comprueba que para obtener una misma probabilidad de error en el bloque determinada, la SNR necesaria es mucho menor para un canal ideal gaussiano que para cualquiera de los otros dos. Podemos comprobar también, como esta SNR es mayor para el canal Pedestrian A que para el Vehicular A, lo

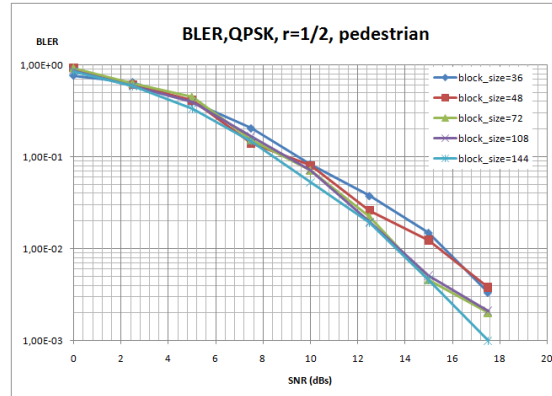


Figura 6.2: BLER para QPSK, tasa 1/2 y canal Pedestrian A

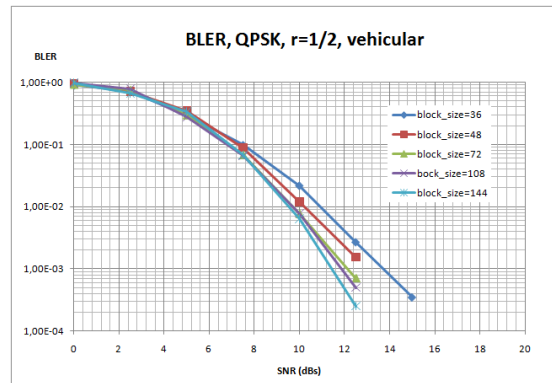


Figura 6.3: BLER para QPSK, tasa 1/2 y canal Vehicular A

que se debe (como ya se ha comentado con anterioridad) a la baja velocidad de movimiento para el canal Pedestrian.

- Puede apreciarse claramente, para los tres canales, cómo el tamaño de bloque elegido para los bloques FEC afecta en el BLER obtenido. Aumentando el tamaño de bloque, se consiguen unas mejores probabilidades de error en el bloque para la misma SNR en cada uno de los canales; obteniendo un ahorro de unos 3 dBs aproximadamente (entre el tamaño más pequeño de bloque y el mayor de ellos) para los canales ITU Pedestrian A y Vehicular A para obtener unas probabilidades aceptables.
- Cuanto mayor es el tamaño de bloque FEC que se utiliza en la transmisión, mayor información (bits) se tiene para que el decodificador iterativo intente corregir los errores producidos en la transmisión. Por esta razón, a mayores tamaños de bloque FEC, mejores probabilidades de error en el bloque se obtienen.
- De las gráficas obtenidas, puede observarse también cómo los resultados en términos de BLER son mejores cuanto mayor es el tamaño de bloque utilizado debido a que se tiende a secuencias codificadas infinitas (como las que Shannon usó al establecer el límite teórico de capacidad del canal)

### 6.3 Efecto del tipo de modulación y de la tasa de código

El segundo de los estudios tiene por objeto el análisis del efecto que la tasa de código elegida tiene en la codificación para las modulaciones QPSK y QAM. En este caso, va a mantenerse constante el tamaño de bloque FEC utilizado y se va a ir variando la tasa de código utilizada para obtener y analizar el efecto que causa en la decodificación en términos de BLER. La tabla 6.2 muestra los parámetros que se han escogido para la obtención de los resultados que serán presentados a continuación.

Tipo de canal	AWGN, Pedestrian A y Vehicular A
Tipo de Modulación	QPSK y QAM
Tasa de código	1/2, 2/3, 3/4 y 5/6
Tamaño de bloque FEC	Fijo
Número iteraciones del decodificador	50

Tabla 6.2: Parámetros de la simulación para el estudio del efecto de la tasa de código

Los datos obtenidos para cada uno de los canales son los que se muestran a continuación:

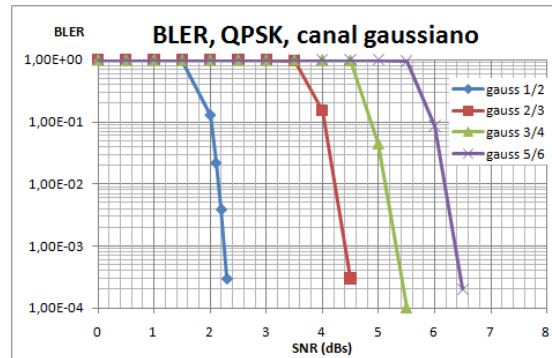


Figura 6.4: BLER para QPSK, FEC fijo y canal AWGN

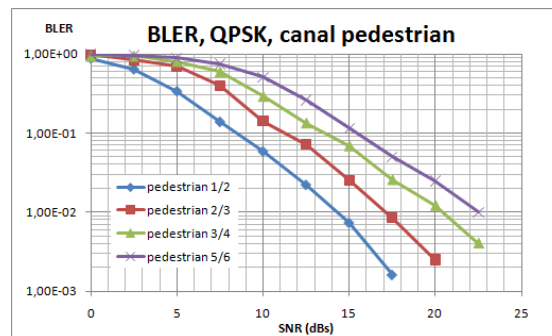


Figura 6.5: BLER para QPSK, FEC fijo y canal Pedestrian A

Como puntos más importantes en este segundo estudio cabe destacar:

- Los resultados de BLER obtenidos para la modulación QPSK son notablemente mejores que para la modulación 16QAM, como era de esperar. Esto es debido a las

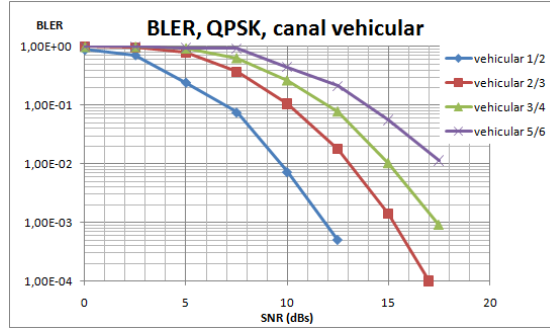


Figura 6.6: BLER para QPSK, FEC fijo y canal Vehicular A

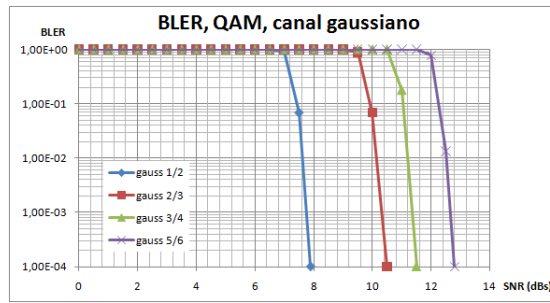


Figura 6.7: BLER para 16QAM, FEC fijo y canal AWGN

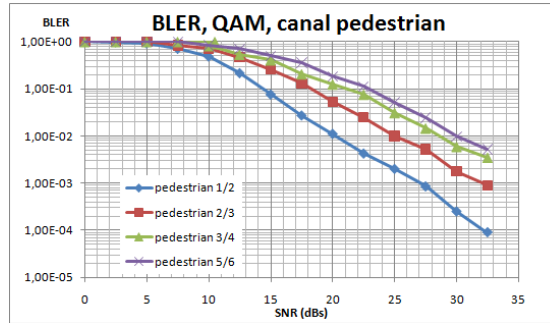


Figura 6.8: BLER para 16QAM, FEC fijo y canal Pedestrian A

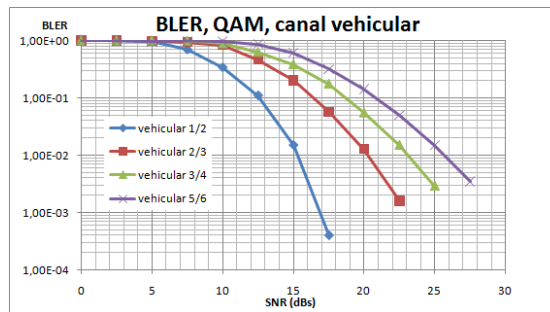


Figura 6.9: BLER para 16QAM, FEC fijo y canal Vehicular A

constelaciones de ambas modulaciones, ya que la constelación QPSK sólo consta de 4 símbolos mientras que la modulación QAM escogida consta de 16; por tanto

la distancia entre símbolos (y su correcta decodificación libre de errores) es menor.

- Puede observarse también cómo la probabilidad de error en el bloque aumenta conforme se disminuye la tasa de código. Esto es así ya que dicha tasa nos indica la proporción de bits que se añaden a los de información (bits de paridad) para la detección y corrección de errores durante la decodificación. Por tanto, la tasa  $1/2$  es la que mayor redundancia nos añade y la que hace nuestra señal más robusta frente al ruido obteniéndose unas probabilidades de error en el bloque muy bajas para SNRs aceptables. En la Figura 6.6, correspondiente a un canal Vehicular A con una velocidad de 120 Km/h, vemos que para obtener una BLER de  $5e-04$  (lo cual es más que aceptable) tan sólo es necesaria una SNR de 12,5 dBs; mientras que para obtener la misma probabilidad de error en el bloque con una tasa de código  $5/6$  necesitaríamos transmitir la información con una SNR superior a los 22 dBs.

## 6.4 Efecto del número máximo de iteraciones permitidas en la decodificación

El propósito de este apartado es analizar el efecto que produce el número máximo de iteraciones que se permiten en la decodificación LDPC iterativa, así como encontrar el número óptimo de iteraciones para alcanzar buenos resultados.

Para este análisis, vamos a utilizar una modulación QPSK con tasa de codificación  $1/2$  y tamaño de bloque FEC fijo; de modo que se va a ir variando el número de iteraciones máximas para ver el efecto que este parámetro causa en el BLER decodificado. La tabla 6.3 nos muestra los parámetros utilizados para este estudio.

Tipo de canal	AWGN, Pedestrian A y Vehicular A
Tipo de Modulación	QPSK
Tasa de código	$1/2$
Tamaño de bloque FEC	Fijo
Número iteraciones del decodificador	2,8,16,32 y 128

Tabla 6.3: Parámetros de la simulación para el estudio del efecto del número máximo de iteraciones permitido

A la vista de los resultados obtenidos y que están representados en las siguientes figuras, podemos sacar las siguientes conclusiones:

- Para los tres canales, y como era de esperar, se observa cómo al aumentar el número de iteraciones el sistema es capaz de obtener resultados de BLER mucho mejores para la misma SNR.
- Puede verse también cómo la mejora que se experimenta al aumentar el número de iteraciones máximas permitidas es cada vez menor, llegando a un punto en el que al aumentar este número máximo de iteraciones permitidas, no se produce ninguna mejora en el BLER obtenido para una misma SNR. Esto delata la

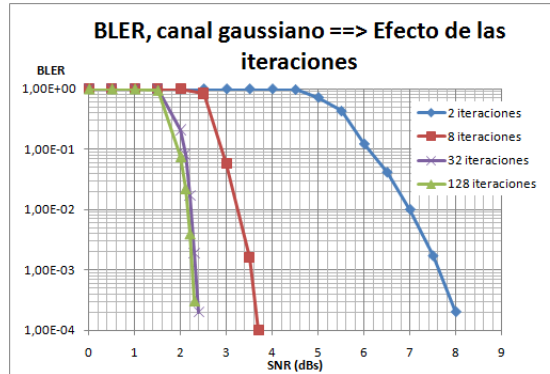


Figura 6.10: BLER para QPSK, FEC fijo y canal AWGN. Efecto del número máximo de iteraciones

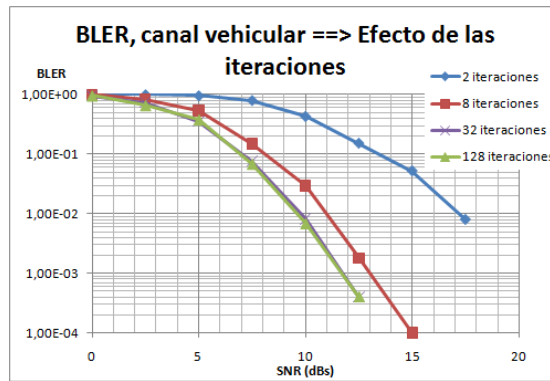


Figura 6.11: BLER para QPSK, FEC fijo y canal Pedestrian A. Efecto del número máximo de iteraciones

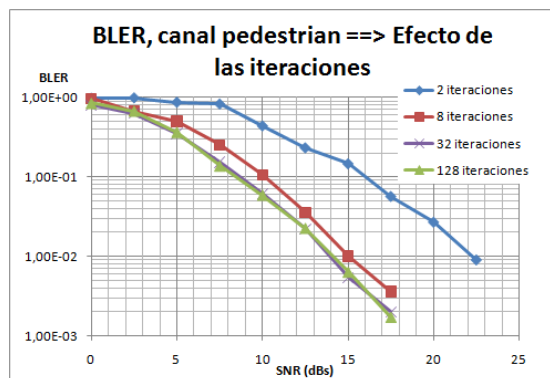


Figura 6.12: BLER para QPSK, FEC fijo y canal Vehicular A. Efecto del número máximo de iteraciones

existencia de un número máximo de iteraciones óptimo, para el cuál se obtienen los mismos resultados en términos de BLER que para valores mayores con una carga computacional menor. Para estos valores de iteraciones máximas permitidas, y con una SNR aceptable, el sistema es capaz de encontrar el bloque decodificado antes de llegar a este número óptimo permitido.

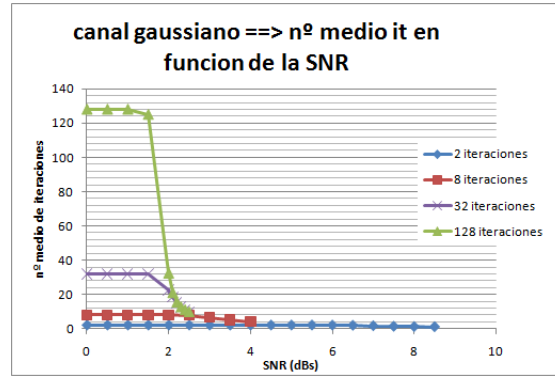


Figura 6.13: Canal AWGN, número medio de iteraciones en función de la SNR

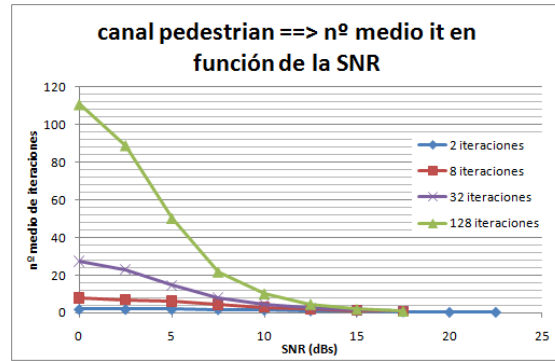


Figura 6.14: Canal Pedestrian A, número medio de iteraciones en función de la SNR

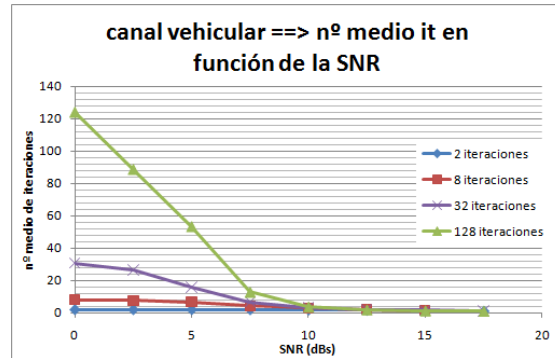


Figura 6.15: Canal Vehicular A, número medio de iteraciones en función de la SNR

En las Figuras 6.10, 6.11 y 6.12 podemos apreciar cómo la respuesta del sistema en términos de BLER mejora notablemente a medida que se aumenta el número máximo de iteraciones permitidas. Se observa también cómo esta mejora va disminuyendo a medida que nos acercamos a un determinado valor de este parámetro, llegando un momento en que el sistema deja de mejorar. Este valor corresponde al valor óptimo que deberá elegirse en el diseño del decodificador bloque LDPC puesto que con él, podemos obtener los mismos resultados que con uno mayor con la carga computacional más baja (ya que para la obtención de los mismos resultados, requiere de la realización de menor número de iteraciones y, por



tanto, menor número de operaciones). Tal como puede apreciarse en las Figuras antes nombradas, y tras las diferentes simulaciones realizadas, encontramos el valor óptimo del número máximo de iteraciones permitidas en 34, puesto que a partir de este valor, la mejoría que se obtiene en términos de BLER es prácticamente inapreciable.

- En las Figuras 6.13, 6.14 y 6.15 puede apreciarse el número medio de iteraciones que el sistema necesita para obtener el bloque decodificado libre de errores. Vemos cómo para relaciones señal a ruido bajas, el sistema llega al número máximo de iteraciones permitido en todos los casos sin que se pueda obtener el bloque decodificado de forma correcta. A medida que la SNR aumenta, el decodificador LDPC necesita menor número de iteraciones para realizar su función puesto que la información viene menos afectada por el ruido. Asimismo, el número medio de iteraciones necesarias para la decodificación tiende a 1 para SNRs altas, puesto que el sistema solo necesita una iteración (en términos medios) para encontrar el bloque FEC decodificado.

## 6.5 Análisis de la carga computacional del decodificador LDPC

A continuación vamos a presentar una tabla con los datos obtenidos en las diferentes simulaciones, donde van a representarse la carga computacional (en términos de operaciones realizadas) que este sistema supone. Estos resultados pueden apreciarse en la tabla 6.4.

SNR	número mult por it	número sumas por it	número medio it	carga computacional mult	carga computacional sumas
2.5	7962624	6635520	26.33	209655889	174713241
5.0	7962624	6635520	15.64	124535439	103779532
7.5	7962624	6635520	6.68	53190328	44325273
10.0	7962624	6635520	3.04	24206377	20171981
12.5	7962624	6635520	1.87	14890107	12408422
15.0	7962624	6635520	1.37	10908795	9090662
17.5	7962624	6635520	1.05	8360755	6967296

Tabla 6.4: Número de operaciones necesarias para la decodificación de un bloque FEC utilizando el decodificador LDPC

La tabla 6.4 muestra el número de operaciones requerido en la decodificación para el peor caso en términos de carga: tamaño de bloque de 144 bytes, tasa de código 1/2. El canal para el que se muestran los datos corresponde al canal Vehicular A.



## Capítulo 7

# Conclusiones

### 7.1 Conclusiones

A lo largo de esta memoria se ha realizado el estudio de la implementación de una codificación bloque LDPC en sistemas WiMAX como alternativa a las técnicas de turbocodificación que actualmente están siendo implementadas en este tipo de sistemas. Se han llevado a cabo tareas de análisis teórico de este tipo de codificadores/decodificadores así como la simulación y presentación de los datos obtenidos con dichos esquemas. Estas simulaciones han tenido por objeto el análisis de todos aquellos parámetros que afectan a este tipo de codificadores para tratar de encontrar el codificador/decodificador LDPC óptimo para el estándar 802.16.

Puesto que en los sistemas WiMAX las condiciones del canal no son constantes; el hecho de elegir parámetros como tipo de modulación, tasa de codificación y tamaño del bloque óptimos no es trivial. Sin embargo, cada servicio ofrecido por el sistema, además de imponer una serie de restricciones (ancho de banda máximo, tasa de transmisión máxima, potencia máxima de transmisión, retardos máximos permitidos, etc), requiere una BLER específica, por lo que tanto la tasa de codificación como el tipo de modulación son seleccionadas con respecto a ella.

A partir de los datos obtenidos en este proyecto, se pueden concluir una serie de aspectos (que serán enumerados a continuación) que ayudarán a la correcta elección de los parámetros WiMAX que mejor se ajusten en cada momento.

Del primero de los estudios realizados, se puede concluir que el aumento en el tamaño del bloque FEC conlleva a obtener mejores resultados en términos de BLER. Con este aumento en el tamaño del bloque se consigue también una mejora de la eficiencia, puesto que se transmiten más bits por portadora con una SNR menor (en comparación con tamaños de bloque menores).

En el segundo de los estudios realizados, se presentó el efecto que tanto el tipo de modulación escogido como la tasa de codificación seleccionada tenía sobre el sistema en cuanto a las probabilidades de error en el bloque que se obtenían. De este estudio, se

puede concluir que cuanto menor es el índice de modulación (en nuestro caso QPSK), mejor es la respuesta del sistema en términos de BLER; por contra, se experimenta un decrecimiento de la eficiencia espectral (menor número de bits por portadora) en comparación con otro tipo de modulaciones. Por esto habrá que establecerse un compromiso entre BLER y eficiencia espectral para la elección del tipo de modulación.

En cuanto a la tasa de codificación, puede decirse que cuanto mayor es ésta, peores probabilidades de error en el bloque se obtienen para una misma SNR. Por el contrario, cuanto menor es la tasa de codificación, mayor es la redundancia que se añade al sistema de modo que se necesitará mayor ancho de banda para transmitir la misma información.

Del último de los estudios puede concluirse que el número de iteraciones máximo que se permite al decodificador influye en el BLER obtenido haciendo que éste sea menor para un número mayor de iteraciones permitido. Sin embargo, cuanto mayor es este número máximo de iteraciones mayor es la carga computacional del sistema puesto que mayor número de sumas y, sobre todo, de multiplicaciones tienen que ser llevadas a cabo. Además de esto, puede concluirse que se aprecia un cierto valor de este número de iteraciones a partir del cual las mejorías del sistema en términos de BLER son imperceptibles y la carga computacional sigue siendo mayor. Por todo ello, habrá que establecer un compromiso entre carga computacional y BLER para escoger el número óptimo de iteraciones.

Por tanto, podemos decir que los bloques LDPC consiguen obtener unos muy buenos resultados en términos de BLER para los tres tipos de canales estudiados con la ventaja de su menor complejidad en la implementación con respecto a los turbocódigos. El mayor inconveniente que este tipo de codificadores bloque tiene, es su elevada carga computacional que se traduce en un elevado número de operaciones (tanto sumas como productos) a ejecutar.

Una vez expuesto todo lo anterior, y para concluir, vamos a enumerar las posibles líneas futuras de trabajo que podrían seguirse tomando como punto de partida el trabajo llevado a cabo en este PFC. Puesto que hemos presentado el coste computacional como uno de los mayores problemas que este esquema de codificación/decodificación tiene, podría usarse este estudio como punto de partida para la implementación de esquemas de codificación LDPC logarítmicas (en las que el principio de codificación-decodificación sigue siendo el mismo pero transforma todas las operaciones a realizar en sumas y restas).

Por otro lado, para conseguir mejores prestaciones en términos de BLER, podría combinarse la implementación del codificador/decodificador LDPC con técnicas de retransmisión HARQ.

# Bibliografía

- [1] R. Van Nee and R. Prasad. “*OFDM for Wireless Multimedia Communications*”. Artech House, 2000.
- [2] C. E. Shannon. “A Mathematical Theory of Communication”. *Bell Systems Technology J.*, 27:379–423, 623–657, julio 1948.
- [3] B. Sklar. “*Digital Communications: Fundamentals and Applications*”. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1988.
- [4] R.G. Gallager. “Low-Density Parity-Check Codes”. Ph.d. thesis, Mass. Inst. Tech., Cambridge, Sept 1960.
- [5] A. Glavieux C. Berrou and P. Thitimajshima. “Near Shannon limit error-correcting coding and decoding: turbo codes”. *Proc. 1993 IEEE International Conference on Communications*, 2:379–423, 1064–1070, May 1993.
- [6] D.J.C. Mackay and R.M. Neal. “Near Shannon limit performance of low density parity check codes”. *Electron Lett.*, 33(6), March 13, 1993.
- [7] D.J.C. Mackay and R.M. Neal. “Good error-correcting codes based on very sparse matrices”. <http://www.inference.phy.cam.ac.uk/mackay/CodesGallager.html>.
- [8] L. M. Tanner. “A recursive approach to low complexity codes”. *IEEE Trans. Inf. Theory*, 25(5).
- [9] Y. Kou S. Lin H. Tang, J. Xu and K. Abdel-Ghaffar. “On algebraic construction of Gallager and circulant low-density parity-check codes”. *IEEE Trans. Inf. Theory*, 50(6).
- [10] A. Trachtenberg T. Etzion and A. Vardy. “Which codes have cycle-free Tanner graphs?”. *IEEE Trans. Inf. Theory*, 45(6).
- [11] Y. Kou S. Lin B. Ammar, J. Xu and B. Honary. “Construction of low-density parity-check codes based on balanced incomplete block designs”. *IEEE Trans. Inf. Theory*, 50(6).
- [12] “IEEE Std 802.16e-2004”. *IEEE Standard for Local and metropolitan area networks*, 2006.

- [13] “IEEE Std 802.16e-2004”. *IEEE Computer Society and IEE Microwave Theory and Techniques Society*, 2004.
- [14] “IEEE Std 802.16e-2005”. *IEEE Computer Society and IEE Microwave Theory and Techniques Society*, 2005.

## Anexo A

# Acrónimos

- **3G:** Tercera Generación
- **4G:** Cuarta Generación
- **ARQ:** Automatic Repeat Request
- **AWGN:** Additive White Gaussian Noise
- **B3G:** Beyond 3G
- **BER:** Bit Error Rate
- **BLER:** Block Error Rate
- **BSID:** Base Station Identifier
- **COFDM:** Coded OFDM
- **CTC:** Convolutional Turbo Code
- **DIUC:** Downlink Interval Usage Code
- **DSL:** Digital Subscriber Line
- **DVB-S2:** Digital Video Broadcasting - Satellite 2
- **DVB-T2:** Digital Video Broadcasting - Terrestrial 2
- **FEC:** Forward Error Correction
- **FFT:** Fast Fourier Transform
- **GSM:** Global System for Mobile communications
- **IFFT:** Inverse Fast Fourier Transform
- **IP:** Internet Protocol
- **IPTV:** Internet Protocol Television

- **ISI:** Intersymbol Interference
- **ITU:** International Telecommunication Union
- **LDPC:** Low-Density Parity-Check
- **LLR:** Log-Likelihood Ratios
- **LSB:** Least Significant Bit
- **LTE:** Long Term Evolution
- **MAN:** Metropolitan Area Network
- **MAP:** Maximum A Posteriori
- **MIMO:** Multiple-Input Multiple-Output
- **MSB:** Most Significant Bit
- **OFDM:** Orthogonal Frequency Division Multiplexing
- **OFDMA:** Orthogonal Frequency Division Multiple Access
- **PFC:** Proyecto Fin de Carrera
- **PRBS:** Pseudo Random Binary Sequence
- **PSK:** Phase Shift Keying
- **QAM:** Quadrature Amplitude Modulation
- **QoS:** Quality of Service
- **QPSK:** Quadrature Phase Shift Keying
- **SNR:** Signal to Noise Ratio
- **TCM:** Trellis Coded Modulation
- **UIUC:** Uplink Interval Usage Code
- **UMTS:** Universal Mobile Telecommunication System
- **VoIP:** Voice over IP
- **Wi-Fi:** Wireless Fidelity
- **WiMAX:** World Wide Interoperability for Microwave Access
- **WWRF:** Wireless World Research Forum