



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**ANÁLISIS DEL DESEMPEÑO DE LA CAPA FÍSICA
BASADA EN OFDM PARA LAS REDES
INALÁMBRICAS DE BANDA ANCHA, BASADAS EN
EL ESTÁNDAR IEEE 802.16-2004**

T E S I S

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN INGENIERÍA
(COMPUTACIÓN)**

P R E S E N T A:

GABRIELA HENRÍQUEZ CAMPOS

DIRECTOR DE LA TESIS: DR. VÍCTOR RANGEL LICEA

MÉXICO, D.F.

2009.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

A mi Padre.

A mi Familia.

A mis Amigas.

Al Dr. Víctor Rangel Licea, asesor de este proyecto.

Al Dr. Rogelio Alcántara Silva.

A los sinodales:

Dr. Gerardo Vega Hernández.

Dr. Javier Gómez Castellanos.

Dr. Miguel López Guerrero.

A la DGAPA por el apoyo recibido del proyecto PAPIIT IN 104907 "Técnicas de Mejoramiento de la capacidad de redes inalámbricas de banda ancha tipo *Mesh*"

RESUMEN

Las redes inalámbricas de banda ancha son la tecnología “de moda” en nuestros días que ofrecen a los usuarios voz, datos y video en alta velocidad. Sin embargo la estandarización de dichas redes aun es un problema. El estándar IEEE 802.16 es la opción de especificaciones estándar para este tipo de redes con mayor aceptación entre las empresas de telecomunicaciones. Actualmente, el *IEEE 802.16 Working Group* ya ha aprobado un estándar para la capa física, el *IEEE 802.16 OFDM PHY Layer*, para la fabricación de equipo, debido al sólido rendimiento en distintos ambientes. En este proyecto se llevan a cabo simulaciones con dos distintos modelos de la capa física. Dado que ambos modelos están diseñados sobre las aplicaciones de *Matlab*, uno en simulación a bloques en “SIMULINK” y el otro programado en *Matlab*, la herramienta que se utiliza para el desarrollo de las simulaciones es *Matlab 7.3*.

El objetivo de estas simulaciones es el poder observar el comportamiento de la capa física basada en OFDM que es descrita en el estándar IEEE 802.16. Dichas simulaciones y sus resultados podrán ser usados como base para futuros proyectos orientados al mejoramiento de los algoritmos de ruteo y QoS.

Los resultados de las simulaciones se muestran en gráficas, donde puede observarse el comportamiento del BER y del BLER para distintos esquemas de modulación y modelos de canal, así como el impacto que tiene el FEC sobre la seguridad de la información.

ÍNDICE

CAPÍTULO I

INTRODUCCIÓN

1.1	Antecedentes	1
1.2	Definición del problema	2
1.3	Objetivos y Metas Particulares	4
1.3.1	Objetivos	4
1.3.2	Metas particulares	4
1.4	Contribución y Relevancia	5
1.5	Estructura de la Tesis	5

CAPÍTULO II

REDES INALÁMBRICAS DE BANDA ANCHA

2.1	Redes Inalámbricas de Banda Ancha	7
2.1.1	802.11 o Wi-Fi	8
2.1.2	802.15 o WPAN	10
2.1.3	IEEE 802.16 o WiMax	11
2.1.3.1	IEEE 802.16a	12
2.1.3.2	IEEE 802.16b	12
2.1.3.3	IEEE 802.16c	12
2.1.3.4	IEEE 802.16d	12
2.1.3.5	IEEE 802.16e	13
2.1.4	IEEE 802.20 Mobile Fi. 16	13
2.1.5	Topología en Redes Inalámbricas de Banda Ancha	14
2.1.5.1	Topología Ad-Hoc	14
2.1.5.2	Topología Infraestructura	15
2.1.5.3	Topología en malla (<i>MESH</i>)	15
2.1.6	Ruteo en Redes Inalámbricas de Banda Ancha	16
2.2	Capas del Protocolo IEEE 802.16 21	17
2.3	La Arquitectura de la Red IEEE 802.16 y su Topología	18

CAPÍTULO III

CAPA FÍSICA

3.1	INTRODUCCIÓN	21
3.2	TIPOS DE INTERFAZ FÍSICA PARA EL IEEE 802.16	22
3.2.1	WirelessMAN-SC	23
3.2.2	WirelessMAN-SCa	25
3.2.3	WirelessMAN-OFDM	26
3.2.4	WirelessMAN-OFDMA	27
3.2.5	WirelessHUMAN	28
3.3	WirelessMAN-OFDM	29
3.3.1	Codificación del Canal	30
3.3.1.1	Randomizer	30
3.3.1.2	FEC	31
3.3.1.2.1	REED-SOLOMON	32
3.3.1.2.2	CÓDIGO CONVOLUCIONAL	32
3.3.1.3	Interleaver	34
3.4	OFDMA	35
3.4.1	Implementación de OFDMA	36
3.4.2	Adición de un Periodo de Guarda para OFDM	37
3.4.3	Consideraciones para el Diseño de un Sistema OFDM	37

CAPÍTULO IV

DESCRIPCIÓN DE LOS MODELOS

4.1	INTRODUCCIÓN	39
4.2	PARÁMETROS DEL SÍMBOLO OFDM	40
4.3	CONFIGURACIÓN DE LA CAPA FÍSICA	41
4.3.1	Randomizer	42
4.3.2	FEC	43
4.3.2.1	Codificador Reed Solomon	43
4.3.2.1.1	Descodificación	44
4.3.2.2	Codificación Convolutacional	44
4.3.2.2.1	Decodificación	45
4.3.2.3	Interleaver	45
4.3.3	Modulación	47

4.3.3.1	Constellation Mapper	47
4.3.3.2	IFFT	47
4.4	MODELADO DEL CANAL	48
4.4.1	Path Loss	48
4.4.2	Retardo de propagación por multitrayectoria (Multipath Delay Spread)	49
4.4.3	Características de desvanecimiento (Fading)	49
4.4.4	Efecto Doppler	50
4.4.5	Modelos de Canales SUI (Stanford University Interim)	50
4.4.6	Implementación de los modelos de canal SUI	52
CAPÍTULO V		
RESULTADOS DE LA SIMULACIÓN		
5.1	Introducción	55
5.2	La simulación	56
5.2.1	BER	57
5.2.2	BLER	61
5.2.3	Efecto del <i>Forward Error Correction</i> (FEC)	62
5.2.4	Efecto de Interleaver	64
5.2.5	Eficiencia Espectral	65
CONCLUSIONES Y FUTUROS PROYECTOS		67
REFERENCIAS		69
ANEXO A		71
CÓDIGO DEL MODELO A BLOQUES DE SIMULINK		
ANEXO B		75
CÓDIGO DEL MODELO COMPUESTO		
ANEXO C		91
GLOSARIO		

ÍNDICE DE TABLAS

TABLA 2.1	REDES BASADAS EN EL ESTÁNDAR IEEE802.11	9
TABLA 2.2	FECHAS IMPORTANTES PARA EL ESTÁNDAR 802.16	11
TABLA 3.1	VARIANTES DE LA CAPA FÍSICA DEL 802.16 Y SUS CARACTERÍSTICAS	22
TABLA 3.2	VALIDACIÓN IEEE 802.16-SC	24
TABLA 3.3	Validación IEEE 802.16-SCa	26
TABLA 3.4	VALIDACIÓN IEEE 802.16-OFDMA	28
TABLA 3.5	CODIFICACIÓN DE CANALES POR MODULACIÓN <i>NG</i>	32
TABLA 3.6	CÓDIGO CONVOLUCIONAL INTERNO CON CONFIGURACIÓN <i>PUNCTURING</i>	33
TABLA 3.7	TAMAÑO DE BLOQUE DEL <i>INTERLEAVER</i> EN BITS	34
TABLA 4.1	PARÁMETROS OFDM O PRIMARIOS	40
TABLA 4.2	PARÁMETROS IEEE802.16 O DERIVADOS	40
TABLA 4.3	CONSTANTES DEPENDIENTES DEL TERRENO.	49
TABLA 4.4	CANAL SUI POR TIPO DE TERRENO	50
TABLA 4.5	CARACTERÍSTICAS GENERALES DE LOS CANALES SUI	50
TABLA 4.6	RETARDO DE PROPAGACIÓN PARA LOS MODELOS DE CABAL SUI	51
TABLA 4.7	TAP POWER (ANTENA OMNI DIRECCIONAL), DE CANALES SUI	51
TABLA 4.8	90 % DE FACTOR K (ANTENA OMNI DIRECCIONAL), DE CANALES SUI	52
TABLA 5.1	ESQUEMAS DE MODULACIÓN Y TASAS DE CODIFICACIÓN SOPORTADOS POR EL ESTÁNDAR 802.16	56

ÍNDICE DE FIGURAS

FIGURA 1.1	DIAGRAMA A BLOQUES DE LAS ETAPAS YA ESTANDARIZADAS	3
FIGURA 2.1	LA FAMILIA 802.11 Y SU RELACIÓN CON EL MODELO OSI	8
FIGURA 2.2	DIAGRAMA DE LAS SUBPORTADORAS PARA OFDM	13
FIGURA 2.3	TOPOLOGÍA AD- HOC	14
FIGURA 2.4	TOPOLOGÍA DE INFRAESTRUCTURA	15
FIGURA 2.5	TOPOLOGÍA DE MALLA (MESH)	15
FIGURA 2.6	PILA DE PROTOCOLOS DEL ESTÁNDAR 802.16	17
FIGURA 2.7	RED IEEE 802.16 TÍPICA	18
FIGURA 2.8	CONEXIÓN PUNTO A PUNTO	18
FIGURA 2.9	CONEXIÓN PUNTO MULTIPUNTO	19
FIGURA 2.10	RED TIPO MESH	19
FIGURA 3.1	TIPOS DE MODULACIÓN	21
FIGURA 3.2	MODOS DE TRANSMISIÓN TDD Y FDD	23
FIGURA 3.3	SUBTRAMA UL PARA WIRELESSMAN SC	23
FIGURA 3.4	TRAMA DL PARA FDD	24
FIGURA 3.5	PROCESO DE TRANSMISIÓN	26
FIGURA 3.6	DESCRIPCIÓN DE LA FRECUENCIA OFDMA	27
FIGURA 3.7	DIAGRAMA CON LAS SUBPORTADORAS OFDM	29
FIGURA 3.8	CADENA DE TRANSMISIÓN PARA LA CAPA FÍSICA OFDM	30
FIGURA 3.9	PRBS PARA ALEATORIZACIÓN DE DATOS	30
FIGURA 3.10	VECTOR DE INICIALIZACIÓN DOWNLINK PARA LA ALEATORIZACIÓN OFDM A PARTIR DE LA RÁFAGA # 2	31
FIGURA 3.11	VECTOR DE INICIALIZACIÓN UPLINK PARA LA ALEATORIZACIÓN OFDM	31
FIGURA 3.12	CODIFICADOR CONVOLUCIONAL DE TAZA 1/2	33
FIGURA 3.13 a)	TÉCNICA MULTIPORTADORA CONVENCIONAL	35
FIGURA 3.13 b)	MODULACIÓN CON PORTADORAS ORTOGONALES	35
FIGURA 3.14	SÍMBOLO OFDM EN EL DOMINIO DEL TIEMPO	35
FIGURA 3.15	DIAGRAMA DE BLOQUES DEL SISTEMA OFDM	36
FIGURA 3.16	INSERCIÓN DEL PREFIJO CÍCLICO	37
FIGURA 4.1	DIAGRAMA A BLOQUES DEL MODELO <i>SENCILLO</i>	41

FIGURA 4.2	DIAGRAMA A BLOQUES DEL MODELO COMPUESTO	41
FIGURA 4.3	DIAGRAMA A BLOQUES DE LA CODIFICACIÓN DEL CANAL	42
FIGURA 4.4	BLOQUE <i>RANDOMIZER</i> DEL MODELO SENCILLO	42
FIGURA 4.5	BLOQUE DE- <i>RANDOMIZER</i> DEL MODELO SENCILLO	43
FIGURA 4.6	BLOQUE "BLOCK ENCODER" DEL MODELO SENCILLO	43
FIGURA 4.7	BLOQUE "BLOCK DECODER" DEL MODELO SENCILLO	44
FIGURA 4.8	BLOQUE "CONVOLUTIONAL ENCOGED" DEL MODELO SENCILLO	44
FIGURA 4.9	EJEMPLO DE DIAGRAMA DE TRELIS PARA UN CODIFICADOR (2,1.3)	45
FIGURA 4.10	BLOQUE "VITERBI DECODER" DEL MODELO SENCILLO	45
FIGURA 4.11	BLOQUE " <i>INTERLEAVER</i> " PARA EL MODELO SENCILLO	46
FIGURA 4.12	BLOQUE " <i>DEINTERLEAVER</i> " PARA EL MODELO SENCILLO	46
FIGURA 4.13	BLOQUE "OFDM SYMBOL" PARA EL MODELO SENCILLO	47
FIGURA 4.14	BLOQUE "OFDM DATA" PARA EL MODELO SENCILLO	47
FIGURA 4.15	ESTRUCTURA GENÉRICA DE LOS MODELOS DE CANAL SUI	51
FIGURA 5.1	PARÁMETROS DE <i>PUNCTURE</i> , CODIFICACIÓN CC 5/6	55
FIGURA 5.2	BER VS SNR USANDO EL MODELO SENCILLO	57
FIGURA 5.3	MODELO A BLOQUES DEL CANAL AWGN DEFINIDO EN EL MODELO SENCILLO	57
FIGURA 5.4	CURVAS BER PARA EL CANAL AWGN DEFINIDO EN EL MODELO COMPUESTO	58
FIGURA 5.5	GRÁFICA DE ESTABILIDAD PARA QPSK $\frac{3}{4}$ USANDO EL MODELO COMPUESTO.	58
FIGURA 5.6	BER VS SNR EN UN CANAL SUI 1	59
FIGURA 5.7	BER VS SNR EN UN CANAL SUI 2	59
FIGURA 5.8	BER VS SNR EN UN CANAL SUI 3	59
FIGURA 5.9	BER VS SNR EN UN CANAL SUI 4	59
FIGURA 5.10	BER VS SNR EN UN CANAL SUI 5	60
FIGURA 5.11	BER VS SNR PARA 64QAM $\frac{2}{3}$ PARA DIFERENTES SUI	60
FIGURA 5.12	BLER VS SNR EN UN CANAL SUI 1	61
FIGURA 5.13	BLER VS SNR EN UN CANAL SUI 2	61
FIGURA 5.14	BLER VS SNR EN UN CANAL SUI 3	61
FIGURA 5.15	BLER VS SNR EN UN CANAL SUI 4	61
FIGURA 5.16	BLER VS SNR EN UN CANAL SUI 5	62
FIGURA 5.17	BLER VS SNR PARA 64QAM $\frac{3}{4}$ PARA DIFERENTES CANALES SUI	62
FIGURA 5.18	EFFECTO DEL FEC SOBRE LA CURVA DE BER PARA QPSK $\frac{1}{2}$ EN SUI2	62

FIGURA 5.19	EFFECTO DEL FEC SOBRE LA CURVA DE BER PARA QPSK $\frac{1}{2}$ EN SUI2	62
FIGURA 5.20	EFFECTO DEL FEC SOBRE LA CURVA DE BER PARA 16QAM $\frac{1}{2}$ EN SUI2	63
FIGURA 5.21	EFFECTO DEL FEC SOBRE LA CURVA DE BLER PARA 16QAM $\frac{1}{2}$ EN SUI2	63
FIGURA 5.22	EFFECTO DEL FEC SOBRE LA CURVA DE BER PARA 64QAM $\frac{1}{2}$ EN SUI2	63
FIGURA 5.23	EFFECTO DEL FEC SOBRE LA CURVA DE BLER PARA 64QAM $\frac{1}{2}$ EN SUI2	63
FIGURA 5.24	EFFECTO DEL <i>INTERLEAVER</i> SOBRE LA CURVA DE BER PARA QPSK $\frac{3}{4}$ EN SUI 3	64
FIGURA 5.25	EFFECTO DEL <i>INTERLEAVER</i> SOBRE LA CURVA DE BLER PARA QPSK $\frac{3}{4}$ EN SUI 3	64
FIGURA 5.26	EFFECTO DEL <i>INTERLEAVER</i> SOBRE LA CURVA DE BER PARA 16QAM $\frac{3}{4}$ EN SUI 3	64
FIGURA 5.27	EFFECTO DEL <i>INTERLEAVER</i> SOBRE LA CURVA DE BLER PARA 16QAM $\frac{3}{4}$ EN SUI 3	64
FIGURA 5.28	EFFECTO DEL <i>INTERLEAVER</i> SOBRE LA CURVA DE BER PARA 64QAM $\frac{3}{4}$ EN SUI 3	65
FIGURA 5.29	EFFECTO DEL <i>INTERLEAVER</i> SOBRE LA CURVA DE BLER PARA 64QAM $\frac{3}{4}$ EN SUI 3	65
FIGURA 5.30	EFICIENCIA ESPECTRAL EN EL CANAL SUI 1	65
FIGURA 5.31	EFICIENCIA ESPECTRAL EN EL CANAL SUI 2	65
FIGURA 5.32	EFICIENCIA ESPECTRAL EN EL CANAL SUI 3	66
FIGURA 5.33	EFICIENCIA ESPECTRAL EN EL CANAL SUI 4	66
FIGURA 5.34	EFICIENCIA ESPECTRAL EN EL CANAL SUI 5	66
FIGURA 5.35	EFICIENCIA ESPECTRAL PARA QPSK $\frac{3}{4}$ EN VARIOS CANALES	66

CAPÍTULO I

INTRODUCCIÓN

1.1 ANTECEDENTES

En un mundo donde el desarrollo depende de la información, el estar comunicado es la base. El principal obstáculo que tienen las comunidades rurales es comunicarse, ya que para ellos las comunicaciones tradicionales (cableadas), además de un alto costo, la mayoría de las veces presentan una imposibilidad geográfica.

Las nuevas tecnologías inalámbricas de banda ancha generan nuevas oportunidades, y al combinarlas con topologías específicas, pueden ser la respuesta al reto de comunicar las regiones más apartadas a un costo aceptable.

Con un ancho de banda de canal superior a los 20 Mbps (**Megabit por segundo**), los sistemas inalámbricos de banda ancha, identificados por las siglas en inglés BWA (*BroadBand Wireless Access*), son la “nueva” tecnología considerada por muchas de las grandes compañías de telecomunicaciones como la mejor opción para superar, tanto en velocidad como en calidad de servicio, a la tecnología actualmente utilizada para proveer servicios de voz y datos.

Los sistemas celulares de tercera generación, 3G, pierden terreno ante todas las características ofertadas por los sistemas BWA que, además de velocidad y calidad de servicio, son en comparación más baratos, lo que trae como consecuencia una reducción significativa al “tiempo de vida” de los sistemas 3G como la tecnología predominante. Algunas de las diferencias que dan ventaja a los sistemas BWA sobre los 3G son las siguientes:

- 3G está enfocado principalmente a la telefonía, mientras que BWA incluye además computadoras portátiles y PDA's (*Personal Digital Assistant*) con acceso de banda ancha móvil.
- Los sistemas BWA están basados en conmutación de paquetes, lo que hace el flujo de información más rápido y optimiza el uso de los recursos de radio. Para 3G la conmutación de paquetes implica económicamente un alto costo y una reducción significativa del ancho de banda.
- La tasa máxima de transmisión ofertada por los sistemas 3G es de 3 Mbps, mientras que los sistemas BWA están diseñados para soportar tasas superiores a los 20 Mbps, lo que supone una diferencia enorme entre ambas tecnologías.
- Las topologías usadas por los sistemas BWA permiten una mayor y mejor cobertura aun en zonas de difícil acceso, como áreas rurales o comunidades aisladas, por ejemplo la topología de malla (*Mesh*).

1.2 DEFINICIÓN DEL PROBLEMA

El estar comunicado en todo momento se ha convertido en una necesidad para la sociedad en general y no únicamente para la comunidad de negocios, lo que ha provocado un rápido crecimiento del mercado de las telecomunicaciones. En diversos artículos sobre las tendencias globales de las telecomunicaciones, se afirma que a finales del 2006 existían alrededor de 4 mil millones de usuarios suscritos a líneas telefónicas, fijas y móviles, y más de mil millones de usuarios de Internet en el mundo. Específicamente existen alrededor de 2700 millones de usuarios suscritos a servicios de telefonía móvil de los cuales el 61% pertenece a países en desarrollo [w1]¹.

En un mercado tan grande y con un crecimiento tan rápido como el de la telefonía móvil es necesario innovar, tanto en el desarrollo de nuevos servicios digitales como en las tecnologías, que permitan proveer de forma más rápida, eficaz y menos costosa a los usuarios de los servicios básicos. Es por ello que las empresas de telecomunicaciones invierten en el desarrollo de nuevas tecnologías para soportar la gran variedad de servicios digitales que ofertan a los usuarios, tales como voz sobre el protocolo de Internet (VoIP), servicios multimedia, videofonía y videoconferencias entre otras aplicaciones.

Los sistemas celulares de tercera generación, 3G, fueron considerados en su momento la alternativa para proveer estos servicios, ya que soportan los sistemas de conmutación de circuitos en los que están basados los estándares de telefonía celular, y los sistemas de conmutación de paquetes necesarios para servicios de transmisión de datos basados en el protocolo IP (*Internet Protocol*). Sin embargo el poder soportar ambas tecnologías resulta costoso y ésta es la razón por la que muchos proveedores de telefonía celular se resisten a implementar los sistemas 3G.

Los sistemas inalámbricos de banda ancha, BWA, basados en sistemas de conmutación de paquetes son considerados actualmente la mejor alternativa para proveer los servicios digitales. Un bajo costo de implementación, aunado a las altas tasas de transmisión de datos y movilidad, son algunas de las características de los sistemas BWA que los ha convertido, para muchas empresas de telecomunicaciones, en la mejor opción.

El protocolo IEEE 802.16 [1] soporta usuarios inalámbricos fijos que no pueden moverse libremente a pesar de tener una conexión inalámbrica, debiendo permanecer fijos en un mismo lugar mientras dure la sesión para no romper la conexión establecida.

La primera versión del estándar IEEE 802.16 [2] funciona en la banda de frecuencias 10-66 GHz y requiere torres con línea de vista (LOS). Posteriormente el estándar modificó su operación con diversas especificaciones para una capa física en la banda de frecuencia de 2-11 GHz, permitiendo las conexiones sin línea de vista (NLOS), que requieren técnicas que compensen eficientemente los efectos de la atenuación y el "*multipath*" (multitrayectoria). Tomando en cuenta las ventajas de la técnica OFDM (*Orthogonal Frequency Division Multiplexing*) para la capa física se puede proporcionar un servicio sólido de banda ancha en un canal inalámbrico hostil.

La capa física basada en OFDM está siendo estandarizada en una estrecha colaboración entre el "*European Telecommunications Standards Institute*" (ETSI) y el "*High Performance Metropolitan Area Network*" (HiperMAN), por lo que la capa física basada en OFDM y el estándar HiperMAN son muy similares. Ambas capas físicas, basadas en OFDM, conforman una técnica OFDM global que aún se está desarrollando. Los productos certificados por el foro WiMAX (*Worldwide Interoperability for Microwave Access*), para BWA son compatibles con ambos estándares.

En lo referente a la capa física basada en OFDM para el estándar IEEE 802.16-2004 [1], existen algunos proyectos que han arrojado como resultado varias tablas comparativas, pero éstas no cubren todas las opciones que la capa física ofrece. Estos datos son relevantes ya que sirven para definir los parámetros de simulación para el resto de las capas del protocolo IEEE 802.16, obteniendo resultados más fiables y apegados a la realidad.

¹ Las referencias a páginas Web se denotarán con una w y el número respectivo, [w1].

De ahí la necesidad de simular todos los esquemas de modulación posibles. La figura 1.1 muestra la pieza central del transmisor y receptor simulados. Estos corresponden a la interfaz aérea de la capa física del IEEE 802.16-2004 basada en *OFDM*.

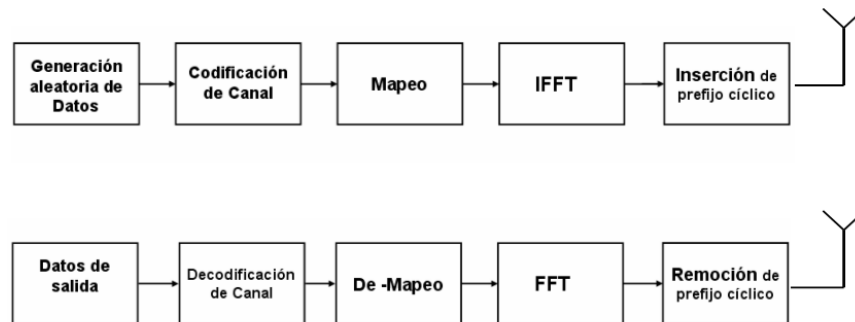


FIGURA 1.1: DIAGRAMA A BLOQUES DE LAS ETAPAS YA ESTANDARIZADAS.

En este esquema sólo se muestran los bloques definidos en la especificación sin incluir los que aun no están estandarizados y que pueden ser objeto de futuros proyectos. El bloque de codificación del canal está compuesto por tres etapas:

- Aleatorización ("*Randomizer*").
- Corrección de errores ("*Forward Error Correction*", FEC).
- Entrelazado ("*Interliving*" o "*Interleaver*").

La corrección de errores se hace en dos fases con un código Reed-Solomon externo (RS) y un código convolucional interno (CC). Las operaciones complementarias se realizan en orden inverso en el canal decodificador para el receptor.

1.3 OBJETIVOS Y METAS PARTICULARES

1.3.1 OBJETIVOS

Realizar una evaluación general de la operación de la capa física utilizando OFDM para las redes inalámbricas de banda ancha, basadas en el estándar IEEE 802.16-2004.

Modelar y simular la capa física del estándar IEEE 802.16-2004 fundamentada en la técnica OFDM usando *Matlab* para tener una mejor comprensión del estándar y del funcionamiento del sistema. Esto implica el estudiar con la simulación los diferentes tipos de modulación de la capa física (BPSK, QPSK, 16QAM, 64QAM), los esquemas de codificación (1/2, 3/2, 3/4 y 5/6) e interpolación en el comportamiento del "bit-error-rate" (BER) y del "block-error-rate" (BLER), bajo los distintos modelos del canal de referencia.

1.3.2 METAS PARTICULARES

La implementación en un ambiente de simulación de laboratorio de un modelo de simulación de la capa física basada en OFDM, usando el software *Matlab* 7.3, que proporcione el comportamiento dinámico de la capa física usando la técnica OFDM en las redes BWA, y que sirva como referencia para poder realizar las siguientes mejoras.

La generación de gráficas con datos confiables sobre el comportamiento de la capa física, que se utilizarán para alimentar las simulaciones del resto de las capas en futuros proyectos.

La publicación de resultados en conferencias, así como la transferencia de las técnicas de optimización que resulten de esta investigación a compañías fabricantes de equipo de redes de acceso inalámbrico de banda ancha; así como también a compañías de telefonía celular para que consideren nuevas opciones hacia la cuarta generación, 4g.

1.4 CONTRIBUCIÓN Y RELEVANCIA

Realizar investigación de vanguardia que permita incrementar y extender el conocimiento actual de las redes inalámbricas de banda ancha. Contribuir al desarrollo de un sistema de información viable, física y económicamente, para localidades apartadas, sentando las bases de un nuevo modelo que cubra las expectativas al respecto.

Esto nos dará una alternativa más para responder al reto de comunicación de las comunidades rurales, con calidad de servicio, seguridad y un costo aceptable.

Evaluar cada uno de los procesos desarrollados en términos de tiempo y calidad de resultados.

Desarrollo de aplicaciones en *Matlab* y gráficas de resultados, que podrán ser usados como base para futuros proyectos orientados al mejoramiento de los algoritmos de ruteo y calidad de servicio (QoS).

1.5 ESTRUCTURA DE LA TESIS

El segundo capítulo se enfoca en la descripción teórica de las redes inalámbricas de banda ancha. Partiendo desde las redes fundamentadas en los estándares IEEE 802.11 hasta las basadas en el estándar IEEE 802.20, se presenta una visión general de sus características particulares.

En el capítulo dos se incluye, también, una breve descripción de la pila de protocolos para las redes fundamentadas en el estándar IEEE 802.16 y de las topologías y arquitecturas que este tipo de redes comparten.

El capítulo tres presenta una descripción más detallada de la capa física, objeto de la investigación básica de este proyecto, y detalles de las diferentes interfaces que pueden implementarse en una red basada en el estándar 802.16. Centrándose al final en la interfaz donde se apoyan los modelos de simulación. Y asimismo una descripción teórica de OFDM.

En el capítulo cuatro se describe la implementación de los modelos, el realizado con bloques de *SIMULINK* [3] y el programado en *Matlab* [4]. Se profundiza en algunas de las etapas descritas en el capítulo tres desde la visión de la implementación.

El capítulo cinco incluye los resultados de simulación y una breve interpretación de los mismos. Así mismo, es donde se dan a conocer algunas de las conclusiones a las que se llegó durante el proceso de esta investigación.

CAPÍTULO II

REDES INALÁMBRICAS DE BANDA ANCHA

2.1 REDES INALÁMBRICAS DE BANDA ANCHA

Las redes inalámbricas no son el futuro de la transmisión de datos, son el presente. El mundo entero se ha convertido en un área de trabajo para aquellos que cuentan con los dispositivos que soportan las diferentes tecnologías inalámbricas (WLAN, Wi-Fi, Bluetooth, 3G, entre otros).

Las tecnologías inalámbricas presentan ciertas dificultades, entre las que destacan la movilidad y la velocidad de transmisión de datos. El problema de la movilidad conlleva principalmente los puntos de acceso, las áreas de cobertura y el efecto que el movimiento tiene sobre la propagación de señales. Para la velocidad de transmisión de datos el problema se traduce como la optimización del ancho de banda.

Las redes inalámbricas surgieron ante la necesidad de un mundo que se mueve por la información, y donde la prioridad es acceder a ella en cualquier momento y en cualquier lugar.

La velocidad de transmisión de datos es la ventaja principal de las redes cableadas, el ancho de banda que se oferta en una de estas redes es de hasta 1 Gbps (**Gigabit por segundo**). Para las redes inalámbricas el ancho de banda se encuentra en el rango de los 50 Mbps, siempre y cuando se cuente con las últimas tecnologías y nos ubiquemos lo más cerca posible del punto de acceso. Existen tecnologías inalámbricas que, en teoría, se encuentran en el rango de los 100 Mbps. Estas últimas son llamadas "Tecnologías Inalámbricas de banda ancha".

La tecnología inalámbrica de banda ancha está cambiando la vida de los usuarios, permitiéndoles conectarse directamente con las personas y acceder a la información mediante una conexión a "alta velocidad", desde cualquier parte, en cualquier instante.

2.1.1 802.11 o Wi-Fi

La familia de protocolos 802 define la tecnología de redes de área local. El estándar IEEE 802.11 o Wi-Fi define características y normas de funcionamiento específicas para los dos niveles más bajos del modelo OSI (*Open Systems Interconnect*), la capa física, PHY y la capa de acceso al medio, MAC, sobre una WLAN (red inalámbrica de área local).

Las principales características de las redes de 802.11 son su simplicidad y robustez contra fallas derivadas del esquema distribuido. La versión 802.11b proporciona velocidades de transmisión de datos de hasta 11 Mbps en un medio inalámbrico, y la nueva versión del 802.11a, utilizando la técnica de modulación OFDM, puede alcanzar velocidades de hasta 54 Mbps en el mismo medio inalámbrico, sobre la banda no licenciada de 5 GHz [5].

Este estándar surgió en 1989 como una forma de proporcionar una red inalámbrica equivalente a Ethernet. Desde entonces se han desarrollado diferentes especificaciones del estándar original con el fin de cubrir los requerimientos de los usuarios. De esta manera surge la familia de estándares IEEE 802.11 [5].

El estándar IEEE 802.11 original, publicado en 1997 y que ofertaba una velocidad de transmisión de datos teórica de hasta 2 Mbps. Las señales infrarrojas (IR) eran el medio de transmisión de datos, el cual aunque sigue formando parte de este estándar actualmente carece de aplicaciones reales.

Este estándar utiliza el protocolo CSMA/CA (*Carrier Sense Multiple Access/ Collision Avoidance*) como método de acceso. En este esquema de codificación una parte significativa de la capacidad de transmisión teórica es utilizada para mejorar la calidad de la transmisión en ambientes adversos, es decir, gran parte de la información enviada es información redundante que se genera por las técnicas de corrección de errores para proteger la información, [6].

En la figura 2.1 se muestra la familia de estándares 802.11.

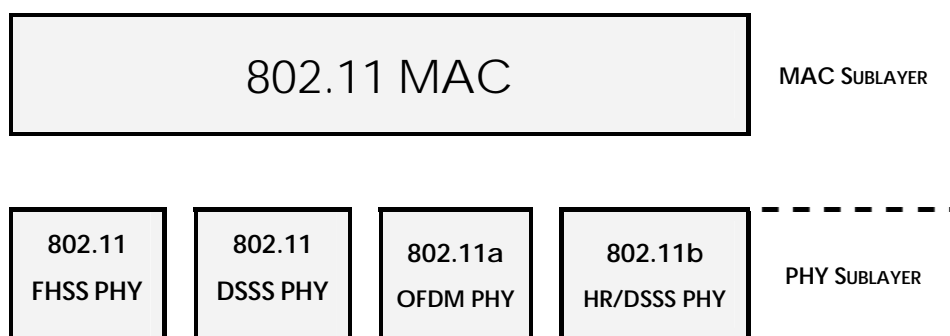


FIGURA 2.1: LA FAMILIA 802.11 Y SU RELACIÓN CON EL MODELO OSI [6].

En la tabla 2.1 se describen brevemente los estándares pertenecientes a la familia 802.11.

ESTÁNDAR	DESCRIPCIÓN
802.11	Publicado en 1997, el estándar original WLAN tolera desde 1 Mbps hasta 2 Mbps. Este estándar utiliza el protocolo CSMA/CA.
802.11a	Aprobado en 1999, este estándar utiliza 8 canales de radio y 52 subportadoras OFDM por canal. Es un estándar WLAN de alta velocidad para la banda de 5 GHz, soporta hasta 54 Mbps.
802.11b	Estándar WLAN para la banda de 2.4 GHz con 3 canales de radio. Aprobado en 1999, el estándar utiliza una modulación conocida como <i>Direct Sequence Spread Spectrum (DSSS)</i> . Tolera hasta 11 Mbps.
802.11c	Estándar que define las características que necesitan los puntos de acceso (Access Points) para actuar como puentes (bridges).
802.11d	Itinerancia (roaming) internacional: configura dispositivos automáticamente para que cumplan con las regulaciones RT locales. Dominios adicionales de regulación.
802.11e	Revisión del estándar en 2005 para garantizar QoS (calidad de servicio), a las Redes Wi-Fi. Dirige la calidad de los requisitos de servicios para todas las interfaces de radio de WLAN IEEE. Posee dos nuevos métodos de coordinación de canal: HCCA (HCF <i>Controlled Channel Access</i>) y EDCA (Enhanced DCF Channel Access).
802.11f	Define comunicaciones del punto de acceso interno para facilitar redes WLAN múltiples distribuidas por proveedores. Inter-Access Point Protocol (IAPP)
802.11g	Ratificado a mediados de 2003 este estándar usa modulación OFDM, pero para hacerlo más compatible con el estándar 802.11b soporta también CCK (<i>Complementary Code Keying</i>) y modulación PBCC (<i>Packet Binary Convolutional Coding</i>). Opera en la banda de frecuencia de 2.4GHz y soporta velocidades de hasta 54 Mbps.
802.11h	Mecanismos de selección dinámica de frecuencias y control de potencia de transmisión para el estándar 802.11a. Define la gestión del espectro de la banda de 5 GHz.
802.11i	Aprobado en Julio de 2004. Dirige las flaquezas de la seguridad actual tanto para los protocolos de codificación como de autenticación. El estándar abarca los protocolos 802.1X, TKIP (<i>Temporal Key Integrity Protocol</i>) y AES (<i>Advanced Encryption Standard</i>).
802.11j	Desarrollado con el objetivo de que los estándares del IEEE (802.11), ETSI (HiperLAN2, <i>High Performance Radio Local Area Network type 2</i>) y ARIB (HISWANA, <i>High-Speed Wireless Local Area Network</i>) puedan coexistir y eventualmente converger.
802.11n	Proporciona mejoras de mayor capacidad de proceso. Se pretende que proporcione velocidades de hasta 500 Mbps. Opera sobre las bandas de 2.4 GHz y 5 GHz, y a diferencia de los anteriores estándares se apoya en un grupo de pequeñas antenas, hasta 4 de ellas, lo que permite ofertar una tasa alta y constante.
802.11m	Mantenimiento para las redes inalámbricas.
802.11X	Métodos de autenticación.

TABLA 2.1: REDES BASADAS EN EL ESTÁNDAR IEEE 802.11.

2.1.2 802.15 o WPAN

Con el fin de desarrollar normas y recomendaciones prácticas para las redes inalámbricas de corto alcance conocidas como redes inalámbricas de área personal (WPAN), en el verano de 1999 el grupo de trabajo IEEE 802.15 se reunió por primera vez. Las WPAN están diseñadas para cubrir las necesidades de los PDA's (*Personal Digital Assistants*), las computadoras personales (periféricos inalámbricos: ratones, bocinas, teclados, entre otros), los teléfonos inalámbricos y algunos sistemas de pago. Las redes WPAN son compatibles con los dispositivos diseñados para funcionar en un espacio personal (POS), esto es en un radio de 5 m alrededor del dispositivo WPAN. El ejemplo de redes WPAN más conocido es el Bluetooth [7].

Las redes WPAN 802.15 operan sobre la banda no comercial de 2.4 GHz usando acceso múltiple por división de tiempo (TDMA, *Time Division Multiple Access*). Este estándar define un enlace de radio de corto alcance, hasta 10 m, y soporta voz o datos a una velocidad máxima de 720 Kbps (Kilobit por segundo), por canal [5].

El grupo de trabajo IEEE 802.15 tiene actualmente los proyectos siguientes:

- **802.15.1:** El objetivo principal de este proyecto es establecer las bases de interoperabilidad y coexistencia entre las redes WPAN 802.15 y las redes WLAN 802.11. Otro de los objetivos es desarrollar normas y especificaciones QoS para diferentes servicios, incluidos voz y datos [5].
Éste es el estándar en el que se apoya Bluetooth.
- **802.15.2:** El objetivo de este proyecto es desarrollar un modelo de coexistencia para cuantificar la interferencia mutua entre una red WPAN y una WLAN, y desarrollar mecanismos de convivencia para facilitar la coexistencia de los dispositivos WPAN y WLAN.
- **802.15.3:** Estándar WPAN *high-rate* (WPAN de alta velocidad). El objetivo es redactar y publicar un estándar WPAN capaz de soportar velocidades de hasta 20 Mbps. También se incluirán soluciones de bajo costo para el estándar anterior [5].
- **802.15.4:** Estándar WPAN *low-rate* (WPAN de baja velocidad). Describe una tecnología simple, de consumo de energía muy bajo, conexiones inalámbricas de bajo costo, y una tasa de transmisión de datos también muy baja. En la versión de 2003 para este estándar en las tasas de transmisión estaban en el rango entre 20 Kbps y 250 Kbps, dependiendo de la banda de frecuencia utilizada.

2.1.3 IEEE 802.16 o WiMAX

El grupo de trabajo IEEE 802.16 desarrolla normas y hace recomendaciones para el desarrollo y despliegue de las redes inalámbricas de banda ancha de área metropolitana (WMAN). El IEEE 802.16 es parte del comité de normas IEEE 802 LAN/MAN (*Local Area Network/ Metropolitan Area Network*). El grupo de trabajo 802.16 se formó en julio de 1999, y en un principio estaba pensado únicamente para redes inalámbricas de banda ancha fijas. Más adelante, el objetivo del comité IEEE 802.16 fue diseñar un sistema de comunicación inalámbrica, que incorpora nuevas tecnologías de comunicación y procesamiento digital de señales para ofrecer Internet de banda ancha, telefonía por Internet (voz sobre IP), y demás servicios, a usuarios móviles en un área metropolitana. En la tabla 2.5 se listan algunas de las fechas importantes para este estándar.

FECHA	EVENTO
Julio 1999	Primera reunión del grupo de trabajo IEEE 802.16.
Junio 2001	Se establece el <i>WiMAX Forum</i> .
Diciembre 2001	Estándar IEEE 802.16 para frecuencias menores a 11GHz terminado.
Enero 2003	Estándar 802.16a terminado.
Junio 2004	Estándar <i>IEEE 802.16 -2004</i> terminado y aprobado.
Septiembre 2004	Intel comienza el envío del primer chip WiMAX.
Diciembre 2005	Estándar <i>IEEE 802.16e</i> terminado y aprobado.
Enero 2006	Primer anuncio de productos certificados por el <i>WiMAX Forum</i> para aplicaciones fijas.
Agosto 2006	Sprint Nextel anuncia sus planes para desplegar WiMAX móvil en los Estados Unidos.

TABLA 2.2: FECHAS IMPORTANTES PARA EL ESTÁNDAR 802.16.

El estándar 802.16 especifica el uso de estaciones base (BS), inalámbricas que están conectadas a redes públicas y estaciones suscriptoras (SS), que proveen el servicio a nivel local a empresas o particulares. Las BS proveen los servicios a las SS.

Originalmente, el grupo de trabajo 802.16 se centró en desarrollar soluciones en las bandas de 10 a 66 GHz, con la función primaria de proveer a las empresas de conexiones de alta velocidad, con un rendimiento superior al que podrían obtener con fibra óptica. El estándar, *WirelessMAN-SC (Single Carrier)*, aprobado en diciembre de 2001, define una capa física que utiliza una modulación de portadora sencilla y una capa MAC con una estructura de multiplexación por división de tiempo (TDM, *Time Division Multiplexing*), que soporta FDD (*Frequency Division Duplexing*) y TDD (*Time Division Duplexing*) [9].

El 802.16a es básicamente una enmienda al estándar original con el fin de incluir aplicaciones de NLOS; el estándar aprobado en diciembre de 2001 requiere LOS, en las bandas de frecuencias de 2 a 11 GHz, usando una capa física basada en OFDM. En el 2004, las revisiones dan lugar a un nuevo estándar llamado IEEE 802.16-2004, conocido también como IEEE 802.16d, que sustituyó a todas las versiones anteriores y estableció la base para la primera solución WiMAX orientada a aplicaciones fijas.

Fue en 2005, con la enmienda conocida como IEEE 802.16-2005 (802.16e), agregó el soporte a la movilidad, dando origen a lo que se conoce como WiMAX Móvil [9].

2.1.3.1 IEEE 802.16a.

Aprobado en enero de 2003 el estándar 802.16a especifica un protocolo que, entre otras cosas, soporta aplicaciones de baja latencia, tales como voz y video. Proporciona conectividad de banda ancha sin necesidad de una línea de vista directa entre las terminales y la estación base. El estándar 802.16a, se diseñó para funcionar en bandas entre 2 y 11 GHz. Un rango tan amplio de bandas de frecuencia es lo que permite al estándar funcionar sin línea de vista.

Entre las aplicaciones que esta especificación agregó al estándar original, como una opción en la capa MAC, es el soporte de OFDMA (*Orthogonal Frequency Division Multiple Access*).

2.1.3.2 IEEE 802.16b.

Anexado al estándar 802.16a, el 802.16b está diseñado para operar en las bandas de frecuencia no licenciadas de 5 a 6 GHz, utilizando anchos de banda por canal de 10 MHz, 20 MHz y 5MHz de forma opcional.

Actualmente este estándar es obsoleto.

2.1.3.3 IEEE 802.16c.

Publicado en enero de 2003, el 802.16c opera en las bandas de frecuencia entre 10 y 66 GHz. Provee niveles mayores de interoperabilidad, protocolos y estructuras *test-suit* (conjunto de pruebas).

El perfil de este estándar se desarrolló para definir cuáles serían las características obligatorias y cuáles serían características opcionales. La meta era orientar a los vendedores sobre los elementos obligatorios que se deben observar para asegurar la interoperabilidad entre los dispositivos.

Al igual que los estándares anteriores, después de la publicación del estándar 802.16-2004 se le consideró obsoleto.

2.1.3.4 IEEE 802.16-2004 o IEEE 802.16d

Aprobado en el verano de 2004, esta versión del estándar agrupó las mejores características de las versiones anteriores y añadió nuevas aplicaciones, haciendo obsoletos los estándares de la familia 802.16 que lo anteceden. También es conocido como WiMAX Fijo.

El estándar 802.16-2004 opera en las bandas 2 a 11GHz, y fue desarrollado para soportar conexiones punto a punto (PTP) y punto a multipunto (PMP).

Este estándar soporta TDD y FDD, además de *Half-duplex* FDD (semi-dúplex FDD). Los primeros perfiles de diseño para WiMAX están basados en TDD debido a las ventajas que ofrece, con excepción de dos perfiles para WiMAX Fijo a 3.5 GHz [9].

El estándar 802.16d define, en su capa física, una interfaz WirelessMAN-OFDM con 256 subportadoras. Estas subportadoras se asignan de la siguiente manera: 192 subportadoras son para datos, 56 son subportadoras nulas usadas como bandas de guarda, 28 al inicio y 27 al final, y una central como frecuencia DC, y 8 subportadoras para pilotos. Ver figura 2.3.

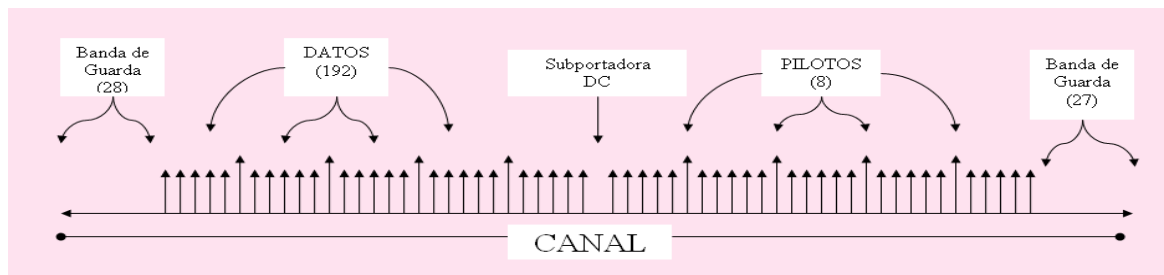


FIGURA 2.2: DIAGRAMA DE LAS SUBPORTADORAS PARA OFDM.

La capa MAC del 802.16d también soporta la topología *Mesh*. La gran diferencia entre PMP y la topología *Mesh* es que en PMP la comunicación se da únicamente entre estaciones suscriptoras (SS) y la estación base (BS), y en la topología *Mesh* las SS pueden comunicarse con otras SS en el vecindario.

2.1.3.5 IEEE 802.16e-2005

El 802.16e, se completó en diciembre de 2005 y se publicó formalmente como IEEE 802.16e-2005. Esta versión del estándar define tres diferentes capas físicas (PHY): portadora sencilla, OFDM, y OFDMA. La técnica de acceso múltiple utilizada en las dos primeras especificaciones de capa física es únicamente TDMA (acceso múltiple por división de tiempo), pero la tercera especificación, OFDMA, ocupa ambas dimensiones, tiempo y frecuencia, para la asignación de recursos. De estas tres tecnologías de capa física, el WiMAX Forum ha seleccionado OFDMA como la tecnología básica necesaria para servicios móviles. Además incluye nuevas modificaciones a la capa MAC para soportar movilidad de alta velocidad. Esta especificación también contempla desplazamiento vehicular en las bandas licencias de 2 a 6 GHz [10].

El estándar 802.16e soporta TDD y FDD. Sin embargo, la versión inicial de los perfiles para WiMAX móvil sólo considera el TDD porque, entre otras características, permite la asignación dinámica de recursos tanto en el canal *Uplink* (UL), como en el canal *Downlink* (DL), lo que permite soportar el tráfico DL/UL asimétrico como el producido por las aplicaciones de Internet. Otra característica del modo TDD es que tanto DL como UL están en el mismo canal de frecuencia y esto provee mayor flexibilidad para la asignación del espectro [10].

Este estándar también es conocido como WiMAX Móvil.

2.1.4 IEEE 802.20 o Mobile-Fi

El estándar IEEE 802.20, aprobado a principios de 2006, está dirigido concretamente a soluciones de banda ancha para movilidad vehicular de hasta 250 Km/h. Este estándar está definido para operar en las bandas por debajo de 3.5 GHz y soportar, en teoría, tasas de datos para el canal DL de 1.2 Mbps, y hasta 4 Mbps en el canal UL. Aun cuando el esfuerzo por desarrollar un estándar similar comenzó hace algunos años, alrededor de 2002, no se ha avanzado mucho, debido a la falta de consenso sobre la tecnología y las cuestiones del proceso de normalización [9].

Este estándar facilita la movilidad al proveer, a los usuarios, una conexión inalámbrica de banda ancha mucho más rápida que cualquiera ofertada a la fecha, y con una cobertura mayor a la de las redes Wi-Fi, lo que ha convertido al estándar 802.20 como la primera opción para suministrar servicios de Internet móvil de banda ancha a los usuarios de los trenes de alta velocidad.

2.1.5 Topología en Redes Inalámbricas de Banda Ancha

En este inciso se describirán brevemente las topologías para las redes WLAN y WPAN para los estándares mencionados anteriormente, la arquitectura y topologías para las redes WMAN se describen más adelante en este capítulo.

2.1.5.1 Topología Ad-Hoc

Esta configuración permite a los usuarios de la red comunicarse directamente entre sí a través de dispositivos inalámbricos sin la necesidad de una infraestructura fija. Esta característica en particular la distingue de otras redes inalámbricas que, como las redes celulares, requieren de una estación base para que los usuarios de las mismas se comuniquen entre sí. Otra de las características de estas redes es que los usuarios pueden conectarse y desconectarse de las mismas con tan sólo encender o apagar sus dispositivos inalámbricos (tarjetas inalámbricas, PDA, celulares, entre otros).

Un ejemplo claro de esta topología son las redes Bluetooth. Dentro de una piconet los dispositivos personales, PDA's, teléfonos celulares o cualquier otro, puede formar parte de una red ingresando al área de cobertura del dispositivo Bluetooth que provee a dicha red, comunicarse con el resto de los miembros de la red sin necesidad de pasar por el dispositivo Bluetooth, y abandonar la red con tan sólo apagar su dispositivo personal o salir del área de cobertura.

En relación con las redes Wi-Fi esta topología presenta las siguientes limitaciones:

1. Los dispositivos de Wi-Fi en modo ad-hoc ofrecen poca seguridad contra conexiones indeseadas. Los intrusos no tendrán dificultad en conectar su dispositivo ad-hoc si consiguen detectar la señal.
2. La potencia de la señal en modo ad-hoc será baja en comparación a la obtenida en el modo de infraestructura. Por lo tanto, al agregar un nuevo componente a una red ad-hoc siempre se tendrán dificultades para mejorar la señal.
3. Los estándares para la implementación de una red Wi-Fi (incluyendo 802.11g) requieren solamente que la comunicación en modo ad-hoc tenga un ancho de banda de 11 Mbps. Se debe considerar que los dispositivos Wi-Fi, que soportan 54 Mbps o más en modo de la infraestructura, alcanzaran un máximo de 11 Mbps cuando están en modo ad-hoc. El modo ad-hoc se considera como "más lento" que el modo de infraestructura por esta razón.

Sin embargo, aun con estas desventajas las redes ad-hoc son superiores en el campo de la movilidad. La figura 2.4 ejemplifica esta topología.



FIGURA 2.3: TOPOLOGÍA AD- HOC.

2.1.5.2 Topología de Infraestructura

En esta topología, la red inalámbrica consta de al menos un punto de acceso (AP, **A**ccess **P**oint), conectado a la infraestructura de una red cableada y un conjunto de estaciones inalámbricas finales. En esta configuración la comunicación entre las estaciones inalámbricas es a través del AP, esto es, si la estación A quiere comunicarse con la estación B el flujo de datos es el siguiente: la estación A se conecta con el AP y el AP se conecta con la estación B, no existe comunicación directa entre las estaciones A y B. El AP es necesario para llevar a cabo una función de puente y conectar varias células WLAN, así como conectar células WLAN a una LAN cableada [11].

La mayoría de las redes WLAN son diseñadas con esta topología, ya que son menos vulnerables que las redes ad-hoc, todos los usuarios deben ser validados por el AP. Al estar conectados a una red cableada la velocidad que pueden ofertar a sus usuarios supera la de una red ad-hoc, y el que toda comunicación entre usuarios deba pasar por el AP garantiza que los paquetes lleguen a su destino. Ver figura 2.5.

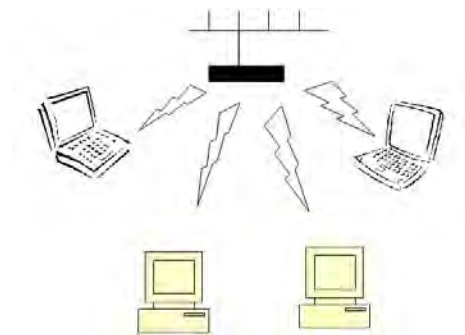


FIGURA 2.4: TOPOLOGÍA DE INFRAESTRUCTURA.

2.1.5.3 Topología en malla (*Mesh*)

La topología en malla es una topología de red parecida a la topología ad-hoc; en el sentido que los nodos pueden comunicarse directamente unos con otros. La diferencia es que en esta configuración los nodos sirven como intermediarios entre sus nodos vecinos y la estación base. Un nodo que esté fuera del alcance de la estación base puede llegar a ella si uno de sus nodos vecinos está dentro de la cobertura de dicha estación base, lo que amplía significativamente el alcance de la red.

Las redes *Mesh* definen dos esquemas de operación: centralizado y descentralizado. El esquema centralizado se refiere a todos los servicios fuera de la célula de cada BS como lo son Internet y telefonía, en este esquema los nodos deben hacer llegar su petición de servicio a la BS. El esquema descentralizado controla los servicios y comunicaciones al interior de la célula, cada nodo actúa como gestor de sus propios recursos. Este tipo de configuración es soportada por las redes WMAN a partir del estándar 802.16. La figura 2.6 representa una topología de malla.

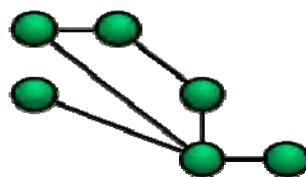


FIGURA 2.5: TOPOLOGÍA DE MALLA (*MESH*) [W2].

2.1.6 Ruteo en Redes Inalámbricas de Banda Ancha

Actualmente la mayoría de las redes inalámbricas son móviles y esto implica una cosa: la estructura de una red inalámbrica varía con el tiempo, tanto en la posición geográfica de los nodos como en el número de los mismos. Esto generó la necesidad de técnicas eficientes para localizar los nodos móviles y asegurar la conexión cuando estos nodos están en movimiento entre dos células distintas (*handoffs*). En el caso especial de las redes de datos, son necesarios protocolos de enrutamiento que tengan en cuenta las topologías dinámicas de las redes.

Tomando el modelo OSI como referencia, se puede ubicar el ruteo (enrutamiento) en la capa de red, cuya principal función es decidir cuál es la mejor ruta para los paquetes. Esta decisión depende de varios factores como la distancia entre origen y destino, el costo de cada ruta en particular, la velocidad de transmisión del canal, la saturación del mismo, entre otros.

Las distancias en las redes se miden en "saltos" (*hops*), un salto por cada nodo o dispositivo intermedio entre el origen y el destino. En una red cableada o de infraestructura se cuenta un salto por cada dispositivo de red por el que el paquete pasa para llegar a su destino. En las redes de ad-hoc o de malla, los saltos son el número de nodos en el vecindario por los que pasa el paquete en su camino.

El costo de una ruta se calcula así: cada enlace entre nodos en una red, ya sea entre sí o con un dispositivo de red, tiene un peso, este peso depende, entre otras cosas, del tipo de enlace y la distancia. El router o la estación base conocen los pesos de cada enlace, en el caso de una red *Mesh* esto aplica también para cada uno de los nodos, y sumando los pesos de cada enlace saben cuál es la mejor ruta para enviar el paquete a su destino.

Para las redes inalámbricas de banda ancha fijas los algoritmos de ruteo pueden ser dinámicos o estáticos.

Los algoritmos estáticos son útiles cuando los nodos definidos son casas, oficinas, edificios y en el caso de las redes inalámbricas de infraestructura. Uno de estos algoritmos es el de "el camino óptimo", que trabaja de la siguiente forma: si un nodo B está en la ruta óptima del nodo A al nodo C, entonces esa ruta contiene la ruta óptima del nodo A al nodo B y la ruta óptima del nodo B al nodo C.

Los algoritmos dinámicos son útiles cuando las redes se componen de nodos menos predecibles como estaciones de trabajo o nodos móviles, ya que este tipo de algoritmo realiza una actualización periódica de las tablas, y mantiene actualizados los cambios en los costos de los enlaces. Ejemplo de estos algoritmos pueden ser los basados en geometría computacional, que ocupan la información geográfica de los nodos, obtenida mediante GPS (*Global Positioning System*).

En algunas redes móviles, como las de malla, se utiliza una técnica dinámica para mantener actualizadas las tablas de la estación base en relación con las rutas que siguen activas. Para esto, la BS manda un mensaje a todos los nodos, que responden enviando sus propias tablas, en función a las cuales la BS actualiza las rutas.

2.2 CAPAS DEL PROTOCOLO IEEE 802.16

La estructura de la pila de protocolos para el estándar 802.16 es casi idéntica a la del resto de las redes que conforman la familia de estándares 802, pero el estándar 802.16 posee más subcapas. La subcapa inferior de la capa física está relacionada con la transmisión. Encima de la capa física de transmisión está una subcapa de convergencia para ocultar las diferentes tecnologías a la capa de enlace de datos [12].

Las versiones *d* y *e* del estándar 802.16 incluyen en la capa MAC una subcapa de convergencia que ofrece funciones específicas para el servicio prestado. Estos servicios incluyen audio / video digital, telefonía digital, ATM (*Asynchronous Transfer Mode*), acceso a Internet y varios más.

La subcapa común MAC está orientada a conexión. Las conexiones en el *Downlink* de la estación base pueden ser *unicast* (un sólo emisor, un sólo receptor), o *multicast* (a varios usuarios), mientras que en el *Uplink* son siempre *unicast*. Para identificar las conexiones se utilizan CID's (identificadores de conexión), de 16 bits.

Las SS revisarán el CID de cada paquete que reciben y sólo conservarán los que van dirigidos específicamente a ellos.

Durante la inicialización de una SS se establecen tres conexiones particulares en ambas direcciones. En estas conexiones se pueden observar tres distintos tipos de requerimientos de QoS usados por diferentes niveles de gestión [13]:

- Conexión básica.
- Conexión de gestión primaria.
- Conexión de transporte.

En la figura 2.7 se encuentra la pila de protocolos para la familia de estándares 802.16, donde se muestran las subcapas específicas de estos estándares.

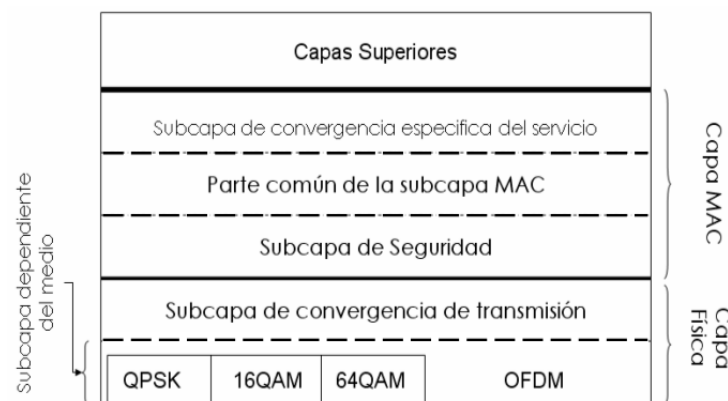


FIGURA 2.6: PILA DE PROTOCOLOS DEL ESTÁNDAR 802.16.

La capa física para la versión *d* del estándar está fundamentada en OFDM y la versión *e* define una capa física basada en OFDMA escalable.

La capa física está definida para un amplio espectro de frecuencias, de 2 a 66 GHz. Para el rango de frecuencias comprendido entre 10 y 66 GHz el sistema requiere LOS. Y en las bandas de 2 a 11 GHz se utiliza propagación NLOS.

En el siguiente capítulo se describe con más detalle los tipos de capa física que están definidos para esta familia de estándares.

2.3 LA ARQUITECTURA DE LA RED IEEE 802.16 Y SU TOPOLOGÍA

Las redes implementadas conforme al estándar IEEE 802.16 presentan una estructura celular. Cada célula cuenta con una estación base, BS y una o más estaciones suscriptoras, SS, dependiendo de la topología implementada. Las transmisiones tienen lugar a través de dos canales: El canal Downlink (de la BS a la SS) y el canal de Uplink, (de la SS a la BS). El canal UL es compartido por todas las SS, mientras que el canal DL es exclusivo de la BS. En la figura 2.8 se muestra una representación de una red 802.16 típica.

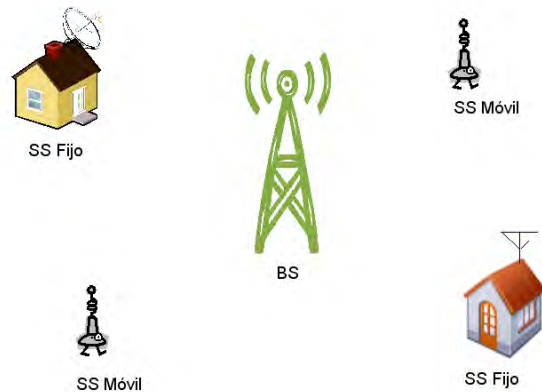


FIGURA 2.7: RED IEEE 802.16 TÍPICA.

Las BS proveen servicios punto a punto, o punto multipunto a varias SS. La configuración punto a punto puede considerarse el equivalente a un enlace dedicado. En las redes donde existe comunicación directa entre nodos, redes ad-hoc, la conexión es PTP. La principal ventaja de este tipo de redes es la descentralización; al permitir conexiones PTP entre los nodos la carga de trabajo de la BS se reduce y puede gestionarse con mayor rapidez las peticiones de servicios externos, requiere LOS. Ver figura 2.9.

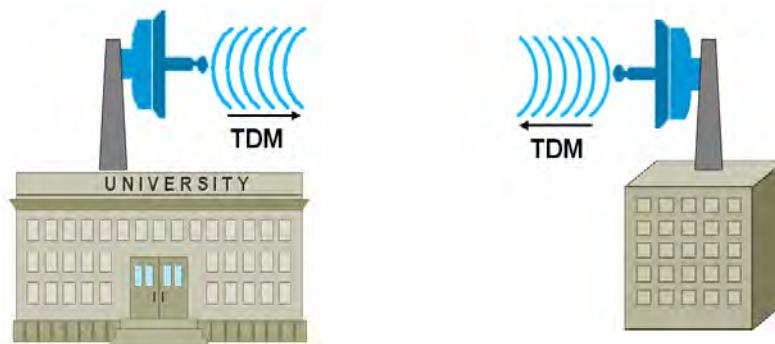


FIGURA 2.8: CONEXIÓN PUNTO A PUNTO [14].

En la configuración punto multipunto, la BS ya no tiene antenas individuales enfocadas a cada nodo de la red. En su lugar tiene una antena de amplia cobertura capaz de abarcar a muchos nodos en su campo de visión. La ventaja de esta configuración es que se simplifica el sistema y se tiene sólo una antena grande en lugar de una infinidad de antenas pequeñas y no requiere LOS. Las desventajas son claras, una antena grande para varios nodos implica compartir los mismos recursos, lo que se traduce en una reducción en la cobertura de la BS y una reducción en las tasas de rendimiento. La figura 2.10 representa una conexión PMP.

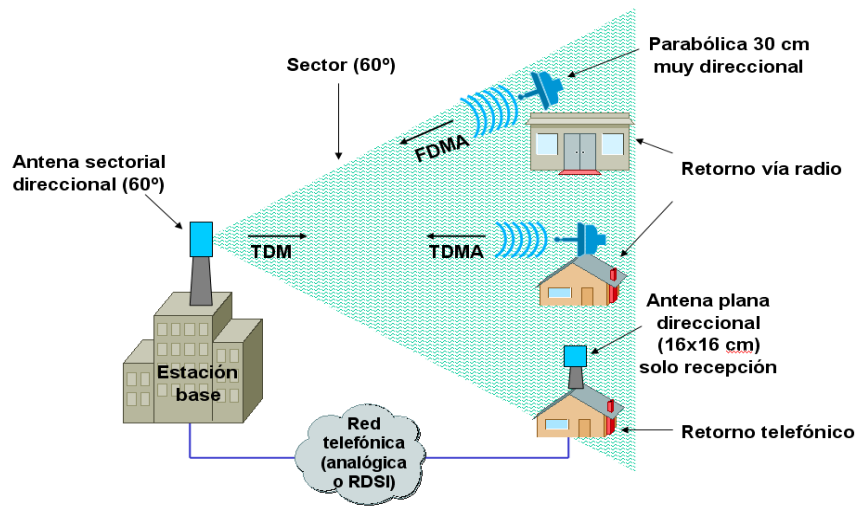


FIGURA 2. 9: CONEXIÓN PUNTO MULTIPUNTO [14].

Existe una tercera topología conocida como *Mesh*, ver figura 2.11. Las redes enmalladas son aquellas en las que la comunicación no sólo se da entre la BS y uno o varios nodos, también existe la comunicación directa entre nodos sin intervención de la BS.

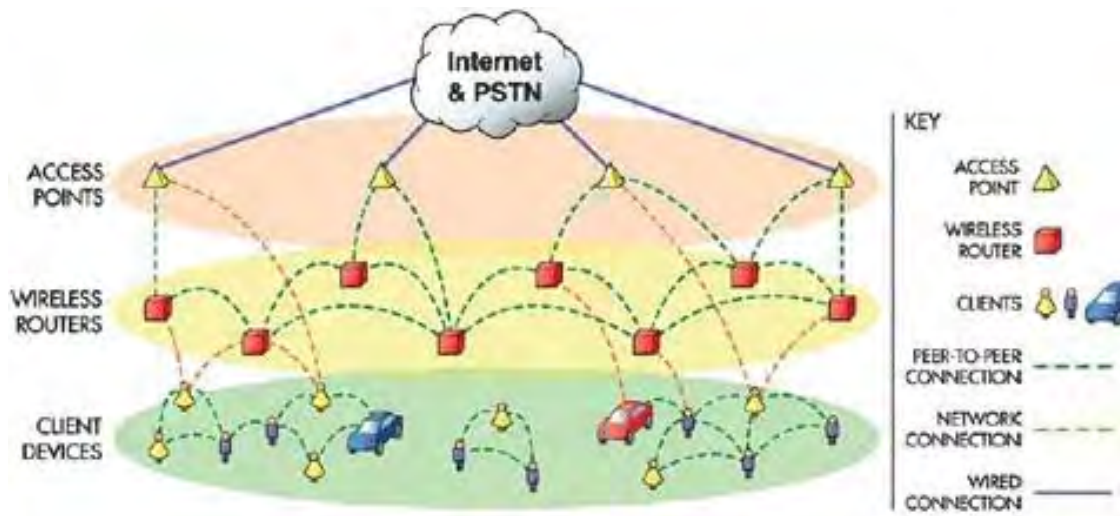


FIGURA 2.10: RED TIPO *MESH* [w3].

Como ya se mencionó en el inciso 2.1.4.3, el esquema de operación al interior de la red es descentralizado y cada SS gestiona sus recursos. La mayor parte del tráfico al interior de la red ocurre sin la intervención de la BS. Cada nodo se comporta como un router, tiene tablas con el identificador de cada nodo en su vecindario y el peso de cada enlace. Sirven como eslabones que permiten al nodo más lejano a la BS comunicarse con ella, lo que amplía el área de cobertura de la BS.

El esquema de operación centralizado está dirigido a gestionar los recursos para los servicios externos, Internet, telefonía celular, entre otros servicios. El gestor de estos recursos es la BS, así que cualquier petición de este tipo debe llegar a ella. La inicialización de un nodo también entra en este esquema.

CAPÍTULO III

LA CAPA FÍSICA

3.1 INTRODUCCIÓN

La capa física es la primera de las siete capas del modelo OSI. Esta capa es la responsable de la transmisión de datos entre los nodos de una red. En una red cableada esta capa define las especificaciones de las señales eléctricas, en una red de fibra óptica define las características de los haces de luz y en las redes inalámbricas la modulación de la señal. El tipo de modulación empleado depende, entre otras cosas, de la distancia entre la estación base y los usuarios. En la figura 3.1 se puede observar una representación gráfica de esto.

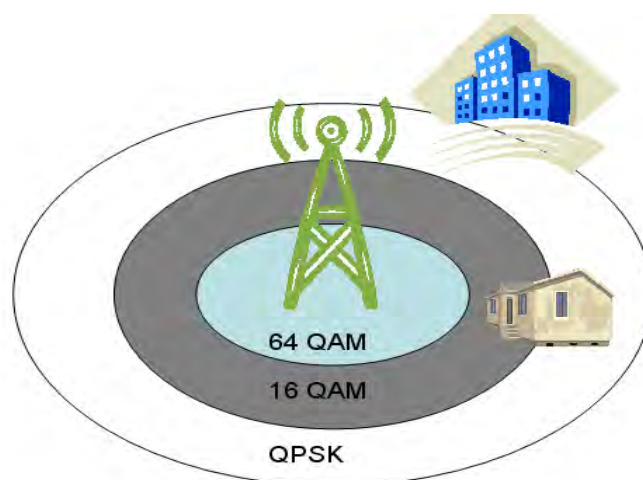


FIGURA 3.1: TIPOS DE MODULACIÓN.

La capa física definida en el estándar IEEE 802.16 original operaba en el rango de frecuencias de 10 a 66 GHz. La versión del estándar 802.16a agregó las bandas en el rango de 2 a 11GHz.

Las diferentes versiones del estándar 802.16 definieron diferentes interfaces de capa física. En este capítulo se dará una breve descripción de estas interfaces y sus principales características.

3.2 TIPOS DE INTERFAZ FÍSICA PARA EL IEEE 802.16

Para las redes WirelessMAN basadas en los estándares 802.16 se definen cinco interfaces aéreas distintas para la capa física.

Estas interfaces son las siguientes:

- **WirelessMAN-SC**
- **WirelessMAN-SCa**
- **WirelessMAN-OFDM**
- **WirelessMAN-OFDMA**
- **WirelessHUMAN**

Las primeras cuatro versiones soportan tanto TDD, como FDD, mientras que la interfaz WirelessHUMAN únicamente soporta TDD. Ambas configuraciones, TDD y FDD, utilizan un mecanismo de ráfagas adaptativas donde los parámetros de transmisión, tales como la modulación y sistemas de codificación, pueden ser modificados para cada SS en particular [1].

Otra de las características es que únicamente la primera interfaz, WirelessMAN-SC, está orientada a conexión LOS. El resto de las interfaces soportan conexiones NLOS.

La tabla 3.1 muestra algunas de las características de estas interfaces aéreas.

CAPA FÍSICA	RANGO	OPCIONES	MODOS DÚPLEX
WIRELESSMAN – SC	10-66 GHz	--	TDD FDD
WIRELESSMAN – SCA	2-11 GHz y bandas licenciadas	AAS ARQ STC	TDD FDD
WIRELESSMAN-OFDM	2-11 GHz y bandas licenciadas	AAS ARQ <i>MESH</i> STC	TDD FDD
WIRELESSMAN-OFDMA	2-11 GHz y bandas licenciadas	AAS ARQ STC	TDD FDD
WIRELESSHUMAN	2-11 GHz y bandas licenciadas	AAS ARQ <i>MESH</i> STC	TDD

TABLA 3.1: VARIANTES DE LA CAPA FÍSICA DEL 802.16 Y SUS CARACTERÍSTICAS.

3.2.1 WirelessMAN-SC

Interfaz de capa física con un esquema de modulación con una portadora única (Single Carrier), opera en las bandas de frecuencia de 10 a 66 GHz. Esta especificación está diseñada para ser flexible, a fin de permitir a los proveedores de servicios optimizar algunos aspectos del sistema, como pueden ser la planificación de la célula, la capacidad del radio, servicios y capacidades [1].

La interfaz de aire WirelessMAN-SC define una arquitectura punto a multipunto, donde la BS básicamente transmite una señal TDM, con cada SS asignado individualmente a una ranura de tiempo (*time slot*), de forma serial.

Esta especificación utiliza un diseño de ráfagas que soporta, como ya se ha mencionado, TDD y FDD. En TDD el UL y el DL comparten un mismo canal, pero no transmiten al mismo tiempo. Para FDD el UL y el DL son canales separados y las transmisiones pueden ser simultáneas. En la figura 3.2 se puede observar una representación gráfica de estos modos de transmisión.

Ambos modos de transmisión, TDD Y FDD soportan ráfagas adaptativas donde los parámetros pueden ser asignados dinámicamente, esto se conoce como un sistema *frame-by frame* (trama a trama).

El enlace ascendente está basado en una combinación de TDMA y DAMA (**Demand Assigned Multiple Access**). En particular, el canal UL se divide en una serie de *time slots*. El número de *time slots* asignados para cada servicio (registro, contención, de guarda o tráfico de usuarios), es controlada por la capa MAC de la BS, y puede variar con el tiempo para optimizar el rendimiento [1].

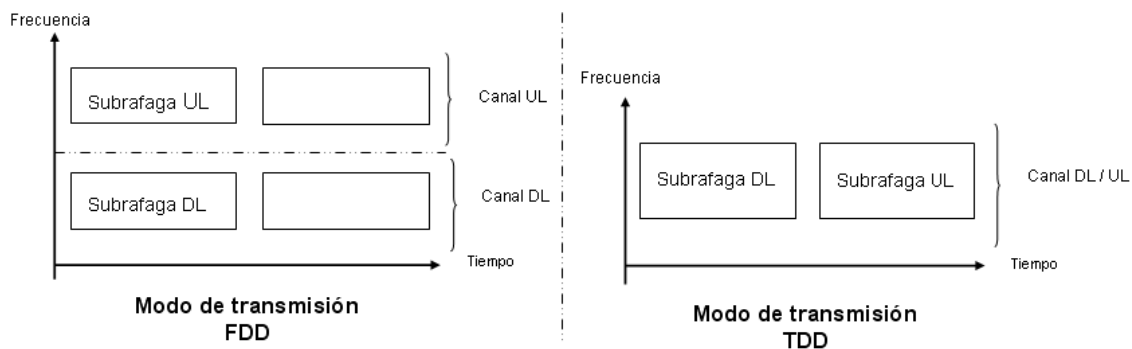


FIGURA 3.2: MODOS DE TRANSMISIÓN TDD Y FDD.

El UL está basado en una ráfaga de transmisión TDMA. Cada ráfaga está diseñada para transportar PDU's (*Protocol Data Unit*) MAC de tamaño variable. En la figura 3.3 se puede observar la estructura de la trama UL.

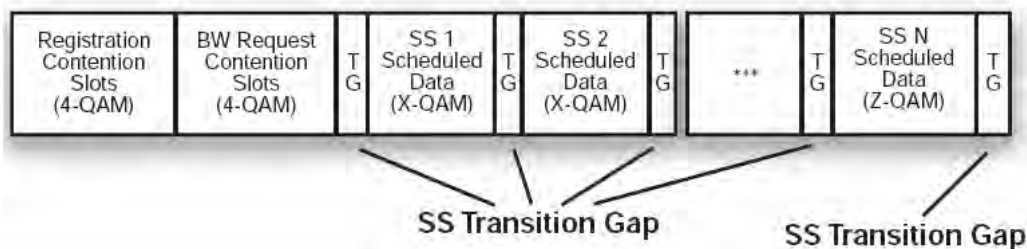


FIGURA 3.3: SUBTRAMA UL PARA WIRELESSMAN-SC [13].

El DL incluye una subcapa de convergencia de transmisión que inserta un byte puntero al inicio del *payload* (carga útil), para ayudar al receptor a identificar el inicio del PDU MAC. El canal DL es TDM, con la información para cada SS multiplexada en un flujo único de datos y recibida por todos los SS en el mismo vecindario [1].

En una ráfaga FDD cada trama DL se subdivide en una serie de ranuras (slots) físicas, y cada ranura contiene cuatro símbolos de modulación. La trama comienza con una sección TDM que está organizada en diferentes grupos de modulaciones y FEC (*Forward Error Correction*). Estos grupos contienen la información transmitida a las estaciones que soportan comunicación bidireccional simultánea (*full-duplex*). La última parte de la trama es la sección TDMA que contiene los datos transmitidos a las estaciones *half-duplex* (envían y reciben pero no a un mismo tiempo) [13].

En la figura 3.4 se muestra la trama para el DL en el modo FDD.

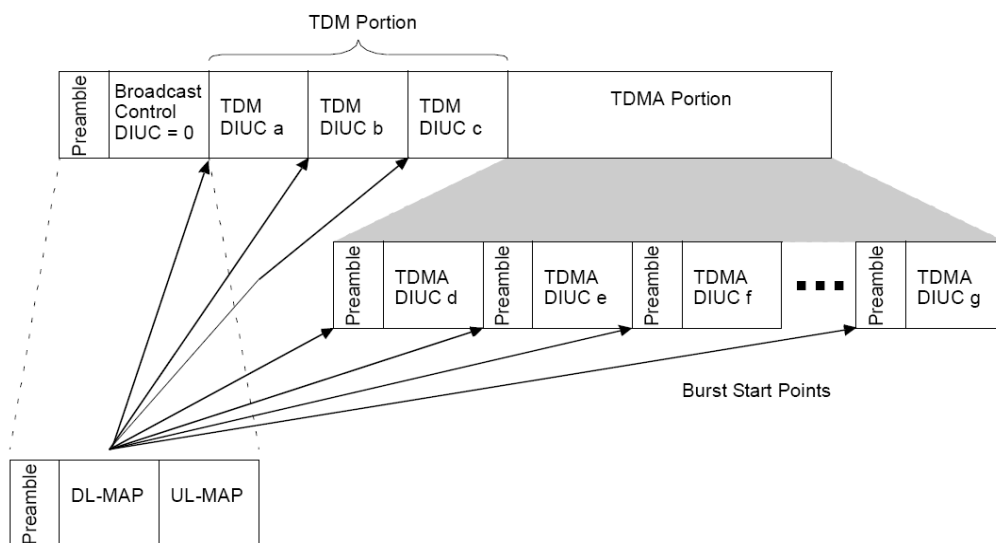


FIGURA 3.4: TRAMA DL PARA FDD [1].

En la tabla 3.2 se encuentran la valoración, el alcance y la velocidad para esta interfaz aérea

CAPA FÍSICA WIRELESSMAN-SC	CONFIGURACIÓN	VELOCIDAD DE CANAL (Mbps)	DISTANCIA DE COBERTURA (Km)			
			ESPACIO LIBRE (LOS)	A	B	C
BW=20 MHz Banda 35 GHz	QPSK EIRP=35dBm	32	1.43	/	/	/
	64-QAM EIRP=35dBm	96	0.23	/	/	/
BW=25 MHz Banda 35 GHz	QPSK EIRP=35dBm	40	1.52	/	/	/
	64-QAM EIRP=35dBm	120	0.24	/	/	/
BW = 28MHz Banda 35 GHz	QPSK EIRP=35dBm	44.8	1.43	/	/	/
	64-QAM EIRP=35dBm	134.4	0.23	/	/	/

TABLA 3.2: VALORACIÓN, ALCANCE Y VELOCIDAD PARA WIRELESS-SC.

3.2.2 WirelessMAN-SCa

La interfaz WirelessMAN-SCa se basa en una tecnología de portadora sencilla, y está diseñada para operación NLOS en las bandas de frecuencia por debajo de los 11GHz. Para las bandas licenciadas, el ancho de banda del canal está limitado por la regulación provisional de anchos de banda, divididos por cualquier potencia de 2, 2^n , superiores a 1.25 MHz [2].

La interfaz WirelessMAN-SCa está definida por cinco diferentes conceptos en la capa física. Estos conceptos son los siguientes:

- TDD
- FDD
- UL TDMA
- DL TDM O TDMA
- Modulación adaptiva por bloque

La capa física incluye codificación FEC, que se utiliza tanto para UL y DL y para las estructuras de trama. Estas funciones ayudan a mejorar la ecualización y la estimación del desempeño del canal en NLOS.

Se definen cinco componentes contenidos en la capa WirelessMAN-SCa [13]:

- **Payload:** Se refiere a las unidades individuales de la transmisión de contenidos (paquetes), que son de interés para alguna entidad en el receptor final.
- **Ráfaga (*Burst*):** Una ráfaga de datos contiene la carga útil y está formada de acuerdo a parámetros definidos por el perfil asociado a ráfaga. La existencia de la ráfaga se da a conocer al receptor a través del contenido de cualquiera de los mapas UL o DL. Para el UL la ráfaga es una unidad completa de transmisión que incluye el preámbulo del encabezado, la carga codificada y la secuencia de terminación.
- **Grupo de Ráfagas (*Burst Set*):** Es una entidad de transmisión autoconcatenada consistente de un preámbulo, una o más ráfagas concatenadas y una secuencia de terminación. Para el UL el grupo de ráfagas es sinónimo de ráfaga.
- **Trama de Ráfaga (*Burst Frame*):** Contiene toda la información incluida en una transmisión simple. Consta de uno o más grupos de ráfagas.
- **Trama MAC (*MAC Frame*):** Hace referencia a los intervalos de ancho de banda fijo, reservados para el intercambio de datos. Para TDD, una trama MAC consiste en una subtrama para el UL y otra para el DL, delimitados por el TTG (*Transmit / Receive Transition Gap*). Para FDD la trama MAC corresponde a la máxima longitud de subtrama de DL. Las subtramas del UL en FDD operan concurrentemente con las subtramas del DL pero en canales de frecuencias separadas.

En el proceso de transmisión los datos fuentes (originales, información), deben ser primero aleatorizados, luego codificados con FEC y mapeados a símbolos QAM (**Q**uadrature **A**mplitude **M**odulation). Los símbolos QAM deben ser entramados dentro de un grupo de ráfagas, lo cual típicamente introduce símbolos adicionales a la trama. Los símbolos dentro de un grupo de ráfagas pueden ser multiplexados en una trama dúplex, que puede contener múltiples ráfagas. Los componentes de símbolo I y Q deben ser inyectados dentro de filtros de pulso de onda, modulados en cuadratura sobre una portadora de frecuencia y amplificados con control de potencia, para que la potencia de salida sea adecuada [2]. En la figura 3.5 se encuentra en el diagrama a bloques que representa el proceso de transmisión.

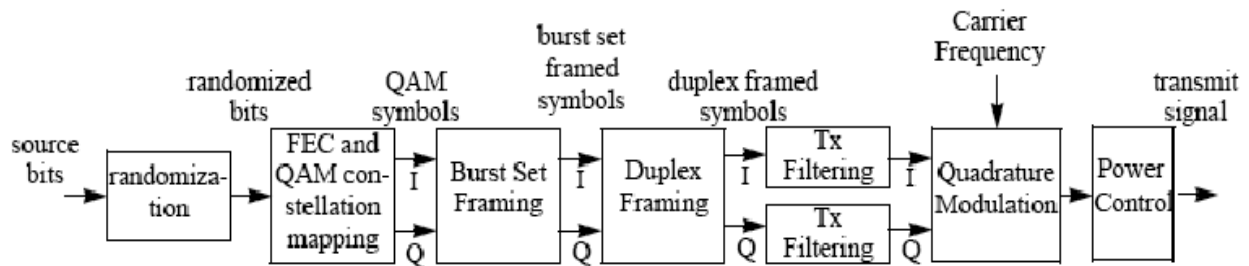


FIGURA 3.5: PROCESO DE TRANSMISIÓN [1].

El proceso de transmisión es el mismo tanto para DL como UL, salvo se especifique lo contrario. En la tabla 3.3 se encuentran la valoración, el alcance y la velocidad para esta interfaz aérea.

CAPA FÍSICA WIRELESSMAN-SCA	CONFIGURACIÓN	VELOCIDAD DE CANAL (Mbps)	DISTANCIA DE COBERTURA (Km)			
			ESPACIO LIBRE (LOS)	A	B	C
BW=3.5 MHz Banda 3.5 GHz	BPSK 1/2 EIRP=35dBm	12.71	26.54	0.68	0.88	1.51
	256QAM 7/8 EIRP=35dBm	17.1	1.88	0.23	0.26	0.42
BW=7 MHz Banda 3.5 GHz	BPSK 1/2 EIRP=35dBm	2.59	18.77	0.59	0.75	1.28
	256QAM 7/8 EIRP=35dBm	36.28	1.33	0.2	0.22	0.35
BW = 10MHz Banda 5.8 GHz	BPSK 1/2 EIRP=35dBm	3.72	10.99	0.45	0.56	0.94
	256QAM 7/8 EIRP=35dBm	52.02	0.78	0.15	0.17	0.26

TABLA 3.3: VALORACIÓN, ALCANCE Y VELOCIDAD PARA WIRELESSMAN-SCa.

3.2.3 WirelessMAN-OFDM

Incluida en el estándar 802.16-2004, la interfaz aérea WirelessMAN-OFDM está fundamentada en un esquema OFDM, que consta de 256 portadoras ortogonales, para operación en modo PMP en condiciones de NLOS, para el rango de frecuencias de 2 a 11GHz.

En el inciso 3.3 se describe un poco más a detalle esta interfaz aérea, ya que es la interfaz definida para los modelos de simulación analizados en este proyecto.

3.2.4 WirelessMAN-OFDMA

La interfaz aérea WirelessMAN-OFDMA usa acceso múltiple por división de frecuencia, con 2048 portadoras y, al igual que WirelessMAN-OFDM, su modo de operación es PMP y no requiere línea de vista (NLOS), en el rango de frecuencias de 2 a 11GHz. Está basada en modulación OFDM.

En la especificación e del estándar 802.16, esta capa física fue modificada a OFDMA escalable (SOFDMA), donde el tamaño para la FFT (*Fast Fourier Transform*), es variable y puede tomar cualquiera de los siguientes valores: 128, 512, 1024 y 2048. El tamaño variable para la FFT permite optimizar la implementación y operación del sistema sobre una amplia gama de anchos de banda del canal y condiciones de radio [9].

Esta capa física tolera subcanalización tanto en UL como en DL. El estándar define cinco tipos de subcanalización. Soporta también los modos de operación TDD y FDD. Tolerancia STC (*Space Time Coding*) y AAS (*Adaptive Antenna System*), con SDMA (*Spatial Division Multiple Access*), un ejemplo de esto es MIMO (*Multiple Input, Multiple Output*). MIMO comprende una serie de técnicas con la utilización de antenas múltiples, tanto en la BS como en las SS, con el fin de incrementar la capacidad y el alcance del canal [13].

En el modo OFDMA las subportadoras activas se dividen en subconjuntos, cada uno de estos subconjuntos es conocido como un subcanal. Un subcanal puede ser asignado a diferentes receptores o grupos de ellos, en el UL a un transmisor se le pueden asignar uno o varios subcanales y varios transmisores pueden operar simultáneamente. Las subportadoras pueden formar un subcanal, pero no deben ser adyacentes [1].

En la figura 3.7 se muestra lo anterior con un ejemplo de un esquema de tres subcanales.

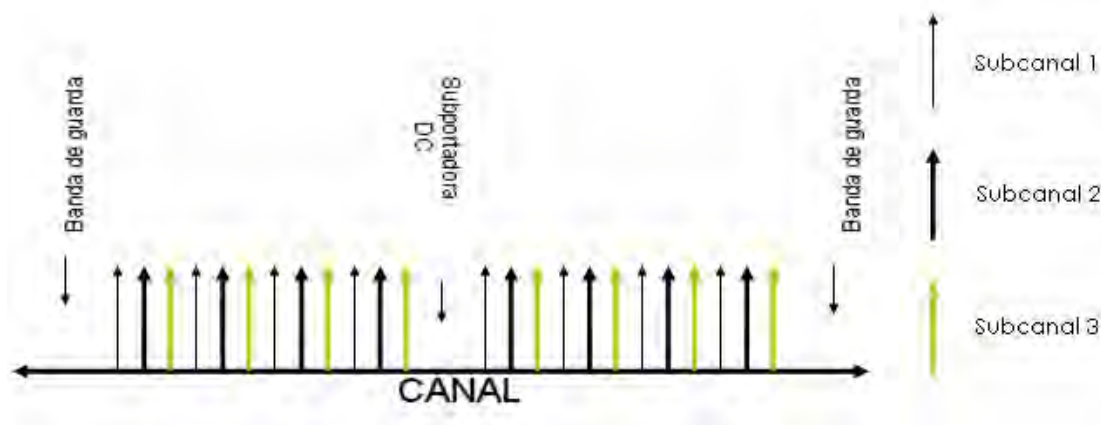


FIGURA 3.6: DESCRIPCIÓN DE LA FRECUENCIA OFDMA.

Al ser una tecnología basada en OFDM comparte muchas de sus características de funcionamiento con la interfaz WirelessMAN-OFDM.

En la tabla 3.4 se encuentran la valoración, el alcance y la velocidad para esta interfaz aérea.

CAPA FÍSICA WIRELESSMAN OFDMA	CONFIGURACIÓN	VELOCIDAD DE CANAL (MBPS)	DISTANCIA DE COBERTURA (KM)			
			ESPACIO LIBRE (LOS)	A	B	C
BW=3.5 MHz Banda 3.5 GHz	64 QAM $\frac{3}{4}$ EIRP=35dBm	12.71	2.73	0.26	0.31	0.5
BW=7 MHz Banda 3.5 GHz	QPSK $\frac{1}{2}$ EIRP=35dBm	5.65	12.19	0.49	0.61	1.04
	64 QAM $\frac{3}{4}$ EIRP=35dBm	24.41	1.93	0.23	0.26	0.42
BW = 20MHz Banda 5.8 GHz	QPSK $\frac{1}{2}$ EIRP=35dBm	16.13	4.8	0.32	0.38	0.63
	64 QAM $\frac{3}{4}$ EIRP=35dBm	72.61	0.76	0.15	0.16	0.26

TABLA 3.4: VALORACIÓN ALCANCE Y VELOCIDAD PARA WIRELESSMAN-OFDMA.

3.2.5 WirelessHUMAN

Llamada "High Speed Unlicensed Metropolitan Area Network", HUMAN, esta interfaz está diseñada para operar en las bandas, exentas de licencia, por debajo de los 11 GHz con canales de 10 y 20 MHz, separados entre sí por 5 MHz.

Opera en sistemas PMP y soporta la topología *Mesh*. Incluye QoS para voz y video en tiempo real.

Al ser una interfaz de capa física fundamentada en OFDM tiene características similares a las capas WirelessMAN-OFDM y WirelessMAN-OFDMA.

3.3 WirelessMAN-OFDM

Esta interfaz usa OFDM con 256 puntos de FFT, lo que le permite operar sin línea de vista. Opera en las bandas entre 2 y 11GHz y es obligatoria para las bandas exentas de licencia. El modo de acceso es TDMA. Ocupada principalmente en redes inalámbricas fijas, donde las SS son puertas de acceso (gateways), ubicados en oficinas y casas.

Esta capa física permite subcanalización en el UL. Dicha subcanalización permite hasta 16 subcanales, con 12 subportadoras por canal que pueden activarse o desactivarse de forma individual.

La interfaz WirelessMAN-OFDM soporta los modos de operación TDD y FDD y permite a las SS operar tanto en modo FDD como *Half-FDD* (semi-FDD). El estándar soporta varios niveles de modulación como son BPSK (*Binary Phase Shift Keying*), QPSK (*Quaternary Phase Shift Keying*), 16QAM, y 64QAM. Incluye además, de forma opcional, diversas transmisiones sobre el DL usando STC (codificación tiempo espacio), y AAS con SDMA (acceso múltiple por división de espacio) [13].

En el esquema OFDM de 256 portadoras estas se distribuyen de la siguiente manera: 192 para los usuarios (datos), 8 para pilotos y 56 para bandas de guarda, de estas 56 una es la subportadora de DC. Ver figura 3.7.

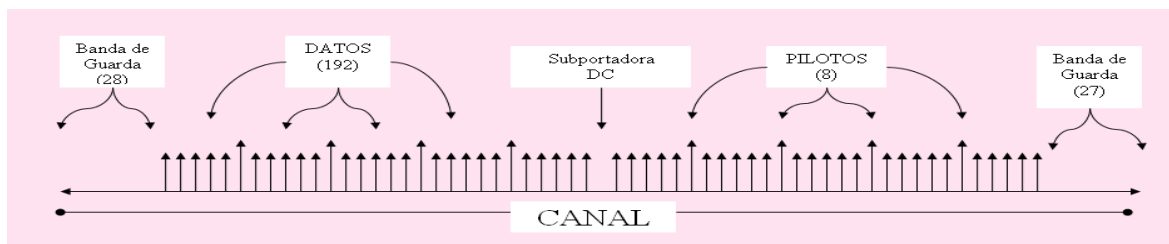


FIGURA 3. 7: DIAGRAMA CON LAS SUBPORTADORAS OFDM.

En esta especificación se definen cuatro diferentes tasas para el CP (prefijo cíclico) que son: 1/4, 1/8, 1/16 y 1/32.

La sincronización en el aire se basa en tramas consecutivas subdivididas en ranuras. El tamaño de cada trama, y el tamaño de cada ranura en la trama, pueden variar trama a trama dependiendo de las necesidades de cada SS.

Las tramas se dividen en subtramas DL y UL. Las subtramas están compuestas por un preámbulo, una trama FCH (*Frame Control Header*), y una serie de ráfagas de datos. La FCH contiene el perfil de la ráfaga y la longitud de una o más ráfagas DL, que siguen a la FCH. El mapa DL (DL-MAP), el mapa UL (UL-MAP), la descripción de los canales DL y UL, DCD y UCD respectivamente, y cualquier otro mensaje que describa el contenido de la trama se envían al principio, en las ranuras iniciales. El resto de las subtramas DL están compuestas por ráfagas de datos para cada SS.

Cada ráfaga está compuesta por un número entero de símbolos OFDM y se le asigna un perfil de ráfaga que especifica el algoritmo de codificación, la tasa de codificación y el tipo de modulación para los datos transmitidos en la ráfaga.

La subtrama UL está compuesta por un periodo de contención inicial, asignación de ancho de banda y PDU's UL de diferentes SS. Los mapas UL-MAP y DL-MAP contienen una descripción completa del contenido de las subtramas UL y DL respectivamente. En ellos se especifica qué SS está transmitiendo y a qué SS está siendo enviada cada ráfaga, el subcanal que cada SS utiliza para transmitir, la tasa de codificación y tipo de modulación para cada ranura y en cada subcanal [13].

En la figura 3.8, se muestra un diagrama a bloques que representa el proceso que los datos recorren, en este caso los datos son PDU's de capa física, antes de llegar al medio físico: aire.

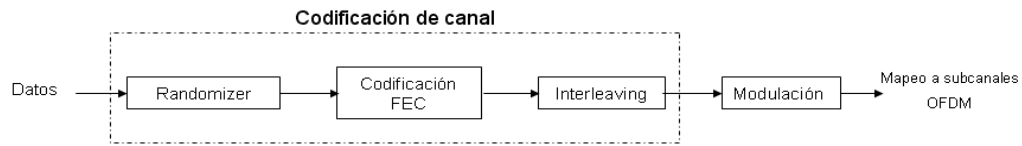


FIGURA 3.8: CADENA DE TRANSMISIÓN PARA LA CAPA FÍSICA OFDM.

3.3.1 Codificación del Canal

Todos los tipos de enlace: cable, fibra óptica, radio, entre otros; son susceptibles a sufrir interferencia de algún tipo. La codificación de canal se desarrolló con el fin de minimizar los daños de dichas interferencias sobre la información. Sus funciones primarias son prevenir y corregir los errores de transmisión. El proceso de codificación del canal en una capa física OFDM se compone de tres pasos: *Randomizer*, FEC e *interleaving*, llevándose a cabo en ese orden en la transmisión y en el orden inverso en la recepción. Ver figura 3.8.

3.3.1.1 *Randomizer*

El *Randomizer* (aleatorización de datos), no es una técnica de corrección de errores, es un mecanismo de prevención de errores implementado en todos los sistemas de transmisión digital. Sirve para conseguir una densidad de potencia más uniforme en el ancho de banda de transmisión para cualquier tipo de portadora digital transmitida, convirtiendo cualquier flujo de bits de información a un número igual de unos y ceros. La aleatorización es necesaria tanto para las BS como las SS.

El *randomizer* se realiza sobre cada ráfaga de datos, tanto en el DL como en el UL. La aleatorización se realiza en el DL y el UL, lo que significa que para un bloque de datos en cada caso (subcanales en el dominio de la frecuencia y símbolos OFDM en el dominio del tiempo), la aleatorización será independiente. Si la cantidad de datos a transmitir no se ajusta exactamente a la cantidad de datos asignados, un relleno (*padding*), de 0xFF (sólo 1's), se añadirá al final del bloque de transmisión. Para la codificación de datos RS-CC o CC el relleno agregado al final del bloque de transmisión podrá ser hasta por la cantidad total de datos asignados menos un byte, esto es, que todo el bloque de transmisión podría ser de relleno con la excepción de un único byte de ceros que introduce el FEC. En el caso de BTC (*Block Turbo Code*) y CTC (*Convolution Turbo Code*), el relleno podría ser hasta por la cantidad total de datos asignados, es decir, el bloque estaría compuesto sólo por el relleno [1].

Cada byte de datos a ser transmitido deberá entrar en forma secuencial en la aleatorización, empezando por los bits más significativos (MSB). El preámbulo está exento de esta etapa. El valor inicial o semilla será usado para calcular los bits aleatorios, que se combinan mediante una operación XOR con el flujo de bits en serie de cada ráfaga. La secuencia de aleatorización sólo se aplica a los bits de información [1]. El generador PRBS (secuencia binaria pseudo-aleatoria) usado para la aleatorización se muestra en la figura 3.9.

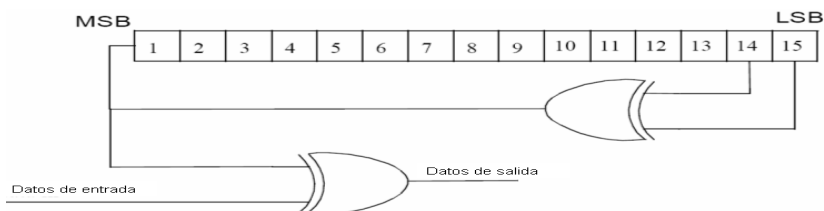


FIGURA 3.9: PRBS PARA ALEATORIZACIÓN DE DATOS [1].

El registro de desplazamiento para la aleatorización se reinicia con cada nueva ráfaga. Para la capa física OFDM, en el DL, comenzará al inicio de cada trama con una secuencia predeterminada (1 0 0 1 0 1 0 1 0 0 0 0 0 0). La aleatorización no se reiniciará al comienzo de la ráfaga # 1. Al inicio de las siguientes tramas, empezando en la trama #2, la secuencia de aleatorización será inicializada como se observa en la figura 3.10. El número de trama usado por la inicialización hace referencia a la trama donde la ráfaga *downlink* es transmitida.

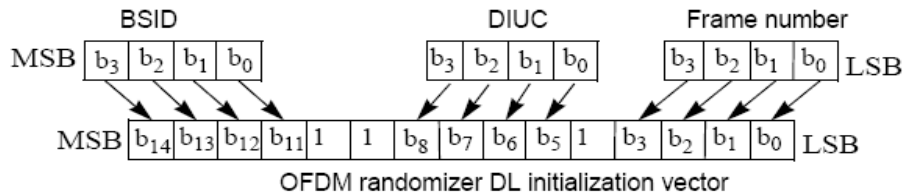


FIGURA 3. 10: VECTOR DE INICIALIZACIÓN DOWNLINK PARA LA ALEATORIZACIÓN OFDM A PARTIR DE LA RÁFAGA # 2 [2].

Para UL, la aleatorización inicializa con el vector mostrado en la figura 3.11. El número de trama usado por la inicialización hace referencia a la trama de la ráfaga uplink que fue transmitida [2].

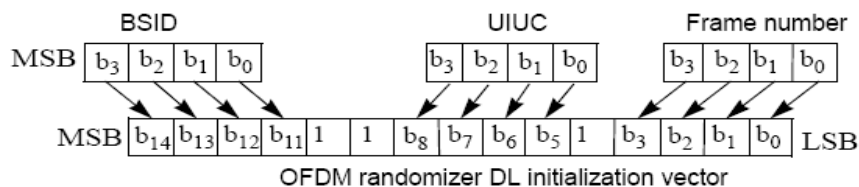


FIGURA 3.11: VECTOR DE INICIALIZACIÓN UPLINK PARA LA ALEATORIZACIÓN OFDM [2].

3.3.1.2 Forward Error Correction o FEC

El FEC o corrección de errores hacia adelante, es un método de corrección de errores que permite al receptor corregir los datos erróneos sin necesidad de una retransmisión de la información original. Se utiliza para sistemas en tiempo real, donde no se puede esperar a la retransmisión para corregir los datos. La corrección de errores se realiza agregando al mensaje original bits de redundancia.

Para la interfaz WirelessMAN-OFDM se define un FEC conformado por la concatenación de un código *Reed-Solomon* (externo), y una tasa compatible de código convolucional (interno). Que es obligatorio tanto en el DL como en el UL. Soporta también, de forma opcional, BTC y CTC. El perfil de ráfaga más completa o, en todo caso, el modo de codificación más sólido debe emplearse cuando se solicite el acceso a la red y en las ráfagas FCH.

Los datos provenientes del *randomizer* son enviados al codificador, que es el encargado de añadir los bits de redundancia, de acuerdo con la tasa de codificación interna, en este caso CC. A la salida del codificador obtenemos una palabra de código, bits de información más bits de redundancia, la relación entre ambos está dada por el codificador externo, *Reed-Solomon*. Esta palabra de código es enviada al receptor y éste, mediante un decodificador y aplicando los algoritmos de corrección de errores, obtendrá la secuencia de datos original.

3.3.1.2.1 REED-SOLOMON

Reed-Solomon es un código cíclico no binario y es también una subclase de los códigos *Bose-Chaudhuri-Hocquenghem* (BCH), que son códigos cíclicos con diversos parámetros, con óptimos resultados para tamaños de bloque de algunos cientos de bits o menos.

Apoyado en las matemáticas de campos finitos, un código Reed-Solomon se define como RS (n, k) con símbolos de s bits. Si $s = 8$, los símbolos son bytes.

El codificador RS opera de la siguiente forma: el codificador toma k bytes de información y añade bytes de paridad para obtener una palabra de código de n bytes. Hay tantos bytes de paridad como $n-k$, es decir, sólo los bytes necesarios para completar la palabra de código. Un decodificador puede corregir hasta t bytes errados por palabra de código, donde $2t = n - k$. Un código RS y sus variables n , k y t pueden escribirse como un vector de la siguiente forma RS ($32, 24, 4$), esto es un código RS con una palabra de código, $n = 32$ bytes, de los cuales son bytes de información $k = 24$, y puede corregir hasta $t = 4$ bytes en error. En la tabla 3.6 se muestran los vectores de RS definidos para esta interfaz.

La tabla 3.5 muestra el tamaño del bloque y las tasas de codificación utilizadas para las diferentes modulaciones y tipos de código.

MODULACIÓN	BLOQUE DE DATOS SIN CODIFICAR (BYTES)	BLOQUE DE DATOS, CODIFICADOS (BYTES)	TASA TOTAL DE CODIFICACIÓN	CÓDIGO RS	TASA DE CODIFICACIÓN CC
BPSK	12	24	1/2	(12,12,0)	1/2
QPSK	24	48	1/2	(32,24,4)	2/3
	36	48	3/4	(40,36,2)	5/6
16QAM	38	96	1/2	(64,48,8)	2/3
	72	96	3/4	(80,72,4)	5/6
64QAM	96	144	2/3	(108,96,6)	3/4
	108	144	3/4	(120,108,6)	5/6

TABLA 3.5: CODIFICACIÓN DE CANALES POR MODULACIÓN.

3.3.1.2.2 CÓDIGO CONVOLUCIONAL

Un código convolucional es un código lineal, donde la suma de dos palabras de código, no importa cuales, es también una palabra de código. Este código se utiliza para mapear k bits de información dentro de una palabra de código de n bits. La implementación de este tipo de código nos da una codificación continua, donde la secuencia de bits codificada depende, además de los bits actuales, de los bits previos.

El código trabaja de la siguiente manera: por cada k bits de información, cuando se codifican se obtienen n bits, donde k/n es la tasa de codificación, por ejemplo en una tasa de codificación de 1/2 por cada bit de información se agrega un bit de redundancia. Esta codificación se realiza usando un registro de desplazamiento, formado por la concatenación de m flip-flops, y una lógica combinacional,¹ implementada por compuertas XOR para realizar la suma módulo 2.

En este codificador se definen las siguientes variables:

k es el número de bits que entra en el codificador,

n es el número de bits a la salida del codificador,

m es la memoria del codificador, es decir el número de flip-flops en el sistema, y

K es la profundidad del codificador y está definido como $m+1$, que es el número de desplazamientos que le toma a un bit entrar y salir del codificador.

En la figura 3.12, está la representación de un codificador convolucional de tasa $\frac{1}{2}$, con los polinomios generadores: $G1=171_{oct}$, para la salida X , y $G2 =133_{oct}$, para la salida Y [1].

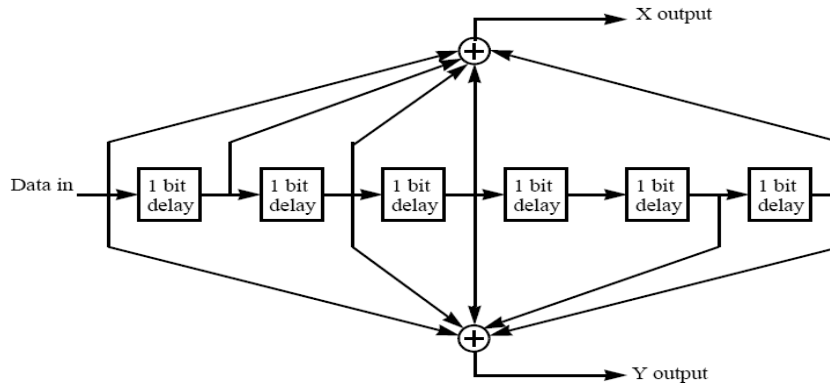


FIGURA 3.12: CODIFICADOR CONVOLUCIONAL DE TASA $\frac{1}{2}$ [1].

En este codificador, entran k bits por segundo y el codificador devuelve $n = 2k$ bits. El bit de entrada se mantiene estable durante el ciclo de codificación, el cual comienza con cada ciclo de reloj. En cada ciclo de reloj, la salida del flip-flop izquierdo entra en el flip-flop derecho, es decir, todos los bits dentro del codificador se recorren una posición y el bit en la entrada del codificador pasa a la salida del primer flip-flop, y un nuevo bit se coloca a la entrada del codificador. Durante el ciclo de reloj el multiplexor de salida se mueve entre las dos posiciones, de tal forma que primero selecciona la salida del sumador superior (la salida X) y después la salida del sumador inferior (la salida Y), generando así el símbolo de dos bits.

Después de la codificación se remueven algunos de los bits de paridad, este proceso es conocido como *puncturing*. El patrón de *puncturing* sirve para adaptar la tasa de los canales de transporte a la velocidad binaria disponible en los canales físicos, en algunos sistemas este patrón es sustituido por un patrón de repetición con el mismo fin. Los patrones de *puncturing* y orden de señalización que son usadas para realizar diferentes tasas de codificación, se definen en la tabla 3.5. En la tabla, "1" significa un bit transmitido y "0" denota un bit retirado, mientras que X e Y hacen referencia a la figura anterior.

	TASA DE CÓDIGO			
Tasa	1/2	2/3	3/4	5/6
d_{free}	4	6	5	4
X	1	10	101	10101
Y	1	11	110	11010
XY	X_1Y_1	$X_1Y_1Y_2$	$X_1Y_1Y_2X_3$	$X_1Y_1Y_2X_3Y_4X_5$

TABLA 3.6: CÓDIGO CONVOLUCIONAL INTERNO CON CONFIGURACIÓN *PUNCTURING*.

Cuando se usa la subcanalización en el UL, el FEC salta la parte de RS y utiliza la tasa de codificación general, definida en la tabla anterior como tasa de codificación CC. El tamaño de bloque no codificado y el tamaño de bloque codificado, pueden ser calculados multiplicando los valores que figuran en la tabla anterior por el número de subcanales asignados dividido por 16 [2], en este caso al usar la configuración por default, 16 canales, los tamaños del bloque sin codificar y el codificado son los de la tabla.

Para la modulación BPSK puede obviarse la parte del codificador RS.

3.3.1.3 Interleaving

El *interleaving*, en español interpolación o entrelazado, es en informática una forma de ordenar datos de una manera no-contigua para mejorar su desempeño. La interpolación es utilizada principalmente en comunicación de datos, transmisión de radio (por ejemplo en satélites) o por ADSL (*Asymmetric Digital Subscriber Line*). El modulo o bloque que realiza este proceso es conocido como *interleaver*. En esta sección se ocupa el término *interleaving* ya que éste es el término utilizado en las revisiones del estándar IEEE 802.16a [2] y IEEE 802.16 -2004 [1], donde se describen los parámetros de esta etapa.

El *interleaving* se utiliza en las tecnologías de transmisión de datos digitales para proteger la transmisión contra ráfagas de errores. Este tipo de errores pueden amontonarse sobre una parte de la transmisión y dañar secciones completas de la información y, como la mayoría de las técnicas de corrección de errores esperan que los errores se distribuyan más uniformemente a largo de la transmisión, este tipo de errores puede resultar abrumador. El *interleaving* ayuda a evitar esto.

Todos los bits de datos codificados son intercalados por un bloque *interleaving*, el tamaño de dicho bloque está dado por el número de bits codificados por subcanal (N_{cbps}), los subcanales asignados y por los bits por símbolo OFDM. El *interleaving* se define por una permutación de dos pasos. La primera asegura que los bits codificados adyacentes sean destinados a subportadoras no adyacentes. La segunda permutación asegura que los bits codificados adyacentes sean destinados, alternativamente, en bits menos o más significativos de la constelación, evitando así que la fiabilidad de los datos baje demasiado.

El *de-interleaving*, que lleva a cabo la operación inversa, también se define por dos permutaciones. La primera permutación en el *de-interleaving* es la inversa de la segunda permutación en el *interleaving*, y viceversa.

En la tabla 3.8 se listan los tamaños para los bloques de *interleaving* de acuerdo con el número de subcanales.

	16 SUBCANALES (POR DEFAULT)	8 SUBCANALES	4 SUBCANALES	2 SUBCANALES	1 SUBCANAL
	N_{cbps}				
BPKS	192	96	48	24	12
QPSK	384	192	96	48	24
16QAM	768	384	192	96	48
64QAM	1152	576	288	144	72

TABLA 3. 7: TAMAÑO DE BLOQUE DEL *INTERLEAVING* EN BITS.

3.4 OFDM

Patentado en 1970 por los laboratorios BELL, OFDM es un método que permite transmitir datos simultáneamente en varias subportadoras. El nombre se debe a que las subportadoras son ortogonales entre sí, lo que significa que ninguna subportadora puede "ver" a las demás subportadoras. Cuando las subportadoras son ortogonales entre sí es posible reducir el ancho de banda total requerido aún más. En la figura 3.13 se muestra gráficamente la diferencia entre una técnica convencional y otra con subportadoras ortogonales.

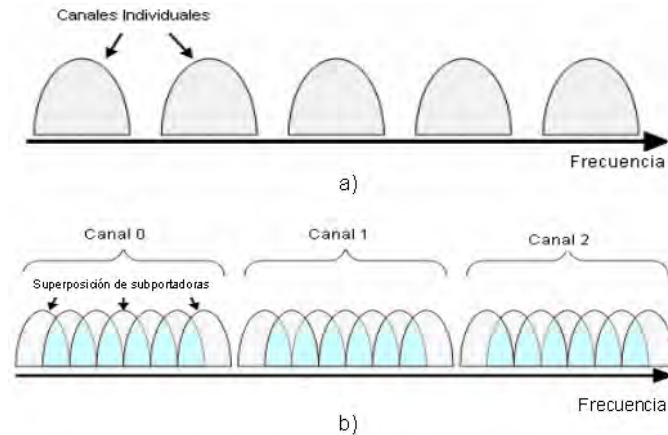


FIGURA 3.13: a) TÉCNICA MULTIPORTADORA CONVENCIONAL, b) MODULACIÓN CON SUBPORTADORAS ORTOGONALES.

El uso de la transformada rápida de Fourier (FFT) y su inversa (IFFT) reducen la complejidad en el receptor, haciendo la implementación de OFDM más sencilla que, por ejemplo, un sistema de portadora sencilla, ya que con la FFT las operaciones por símbolo OFDM son del orden de $N \log(N)$. OFDM es también resistente a los desvanecimientos de frecuencia y a las interferencias. Con información del estado del canal, puede implementarse efectivamente la detección de probabilidad máxima de errores para cualquier perfil de canal o patrón de interferencia.

OFDM presenta algunas desventajas como lo es su susceptibilidad a los errores de la frecuencia portadora y a la interferencia intercanal (ICI).

En la figura 3.14 se muestra la estructura del símbolo OFDM en el dominio del tiempo, de acuerdo con el estándar IEEE 802.16-2004.

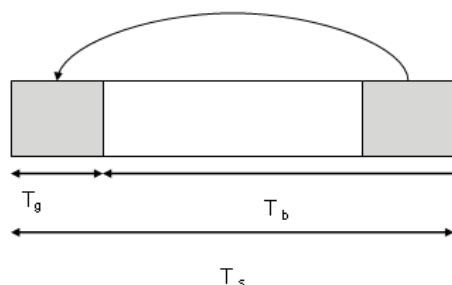


FIGURA 3.14: SÍMBOLO OFDM EN EL DOMINIO DEL TIEMPO [1].

El estándar IEEE 802.16-2004 define una serie de parámetros para caracterizar a los símbolos OFDM. Estos parámetros se dividen en primarios y derivados. Los parámetros primarios son:

BW Es el ancho de banda nominal del canal.

N_{used} Número de subportadoras utilizadas.

n Factor de muestreo. Este parámetro, en conjunto con BW y N_{used} determina la distancia entre subportadoras, y el tiempo útil del símbolo.

G Es la razón entre el tiempo del prefijo cíclico (CP) y el tiempo útil del símbolo.

Los parámetros derivados son:

$NFFT$ La potencia de 2 más pequeña, pero mayor que N_{used} .

F_s Frecuencia de muestreo. $F_s = \text{floor}(n \cdot BW/8000) \times 8000$

Δf Espacio entre subportadoras. $\Delta f = F_s / NFFT$

T_b Tiempo útil del símbolo. $T_b = 1 / \Delta f$

T_g Tiempo del prefijo cíclico (CP). $T_g = G \cdot T_b$

T_s Tiempo del símbolo OFDM. $T_s = T_b + T_g$

$T_{sampling}$ Tiempo de muestreo. $T_{sampling} = T_b / NFFT$

3.4.1 Implementación de OFDM

La implementación del sistema digital OFDM se logra a través de las operaciones matemáticas llamadas transformada discreta de Fourier (DFT) y transformada discreta inversa de Fourier (IDFT). Estas dos operaciones son ampliamente utilizadas para la transformación de datos en el dominio de tiempo y el dominio de frecuencia. En el caso de OFDM, estas transformadas pueden considerarse como datos cartográficos de las subportadoras ortogonales. En la práctica, los sistemas OFDM utilizan bloques de FFT e IFFT, que son la versión matemática equivalente de DFT e IDFT respectivamente.

En la figura 3.15 se muestra el diagrama a bloques de un sistema OFDM.

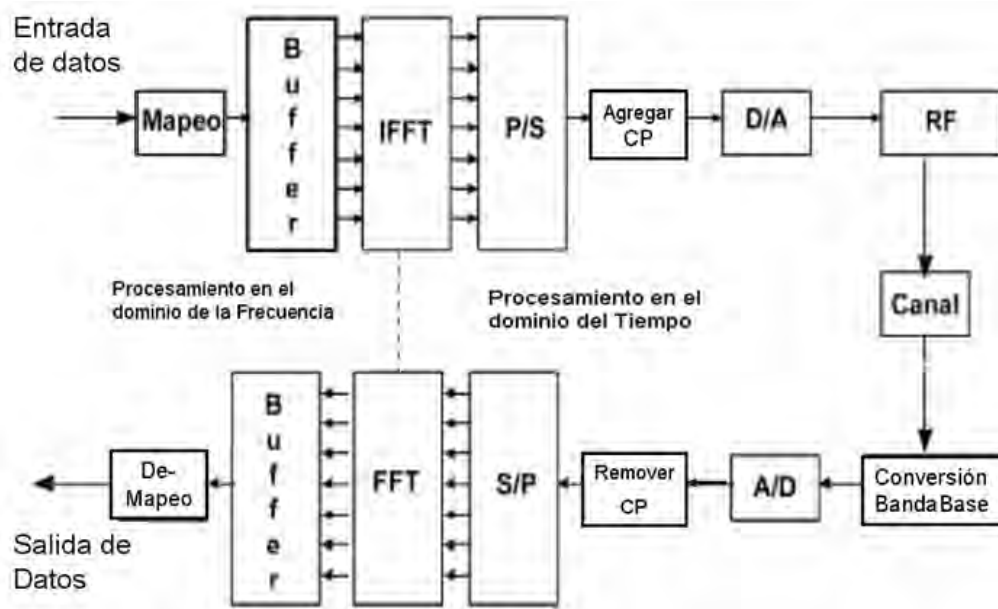


FIGURA 3.15: DIAGRAMA DE BLOQUES DEL SISTEMA OFDM.

En un sistema de OFDM, en la etapa de transmisión, se considera que los datos fuente están en el dominio de la frecuencia. Estos datos se introducen a un bloque IFFT que lleva la señal al dominio del tiempo. El sistema divide el flujo de datos en N flujos paralelos, donde N es una potencia de 2, y cada uno de estos flujos es mapeado a una subportadora. La base para las funciones del bloque IFFT está constituida por N sinusoides ortogonales, una por subportadora, cada una de las cuales posee una frecuencia diferente, desde 0 (DC) hasta el valor máximo soportado por la "banda base", y la IFFT recibe N símbolos a la vez. Cada N valores complejos de los símbolos de entrada determinan la amplitud y fase de la senoide para cada portadora. A la salida de la IFFT se obtiene una poliarmónica en formato discreto, que es una suma de las N sinusoides, conocida como símbolo OFDM. La longitud del símbolo OFDM es NT donde T es el período del símbolo de entrada de la IFFT. De esta forma, el bloque IFFT provee una manera de modular los datos sobre N subportadoras ortogonales.

Después de la modulación OFDM, los datos son convertidos, nuevamente, a una forma serial y se agrega un intervalo de guarda a cada símbolo OFDM.

La recepción es el proceso inverso, es decir se realiza la conversión de los datos del dominio del tiempo al dominio de la frecuencia.

3.4.2 Adición de un Periodo de Guarda para OFDM

Una de las propiedades más importantes de las transmisiones OFDM es la solidez en contra de los retardos de propagación por multitrayectoria. Esto se logra aumentando el período del símbolo, lo que minimiza las interferencias Inter-símbolo. De hecho, el nivel de robustez puede aumentar aún más mediante la adición de un período de guarda comprendido entre los símbolos transmitidos. El período de guarda da tiempo a que las múltiples señales de los símbolos previos se desvanezcan antes de que la información del símbolo actual se una. La forma más eficaz de periodo de guarda usado es una extensión cíclica del símbolo. Si, en el dominio del tiempo, se agrega una copia espejo del final de la onda del símbolo al comienzo del mismo, como período de guarda, ésta extiende efectivamente la longitud del símbolo, al tiempo que se mantiene la ortogonalidad de la forma de onda. En la figura 3.16 puede observarse una representación gráfica de la anterior. El uso de este símbolo cíclico amplía las muestras necesarias para el desempeño de la FFT (para decodificar el símbolo), se pueden tomar en cualquier lugar a lo largo del símbolo.

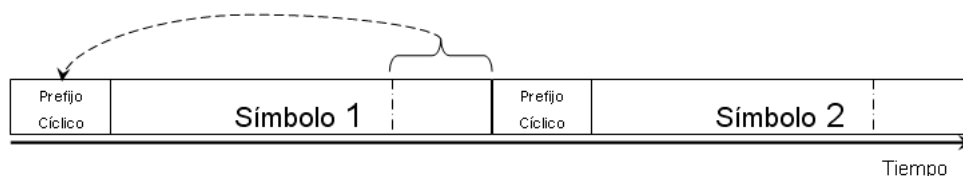


FIGURA 3.16: INSERCIÓN DEL PREFIJO CÍCLICO.

3.4.3 Consideraciones para el Diseño de un Sistema OFDM

El objetivo que se persigue al diseñar un sistema basado en OFDM, y en general cualquier otro sistema de comunicación, es garantizar la información. Para ello, en este caso, podemos reducir la tasa de transmisión de datos de las subportadoras, lo que aumentaría el tiempo por símbolo y, como resultado, reduciría la interferencia. El agregar un CP alto también daría buenos resultados contra la interferencia, sin embargo esto generaría otros problemas. Por tanto se debe buscar un equilibrio para maximizar las ventajas y reducir los efectos secundarios.

El sistema OFDM depende de cuatro requerimientos básicos, que son:

- **Ancho de Banda Disponible:** El ancho de banda es siempre un recurso escaso, por lo que la base del diseño del sistema debe ser la disponibilidad de ancho de banda para la operación. La cantidad de ancho de banda desempeña un papel importante para determinar el número de subportadoras, ya que, con un gran ancho de banda, se puede acoplar fácilmente un gran número de subportadoras con un espacio de guarda razonable.
- **Tasa de Bits:** El sistema debe ser capaz de tolerar la velocidad de transmisión de datos requerida por los usuarios, o la necesaria para soportar los servicios ofertados.
- **Tolerancia al Retardo de Propagación:** Lo que se define como un retardo de propagación tolerable depende del medio ambiente del usuario. Las mediciones muestran que en ambientes interiores se experimenta un retardo de propagación máximo de algunos cientos de nano-segundos, mientras que en ambientes al aire libre puede ser de hasta 10 μ s. De modo que la duración del CP debe determinarse de acuerdo con la tolerancia al retardo de propagación.
- **Valores Doppler:** Los usuarios de un vehículo a alta velocidad experimentarán un mayor efecto *Doppler*, mientras que los peatones experimentarán un menor efecto *Doppler*. Estas consideraciones deben tenerse en cuenta.

Los parámetros de diseño se derivan de acuerdo con los mismos requisitos. Los parámetros de diseño del sistema deben satisfacer los requisitos para el diseño del sistema. Los siguientes son los parámetros de diseño para un sistema de OFDM.

- **Número de Subportadoras:** El aumento del número de subportadoras reducirá la tasa de transmisión de datos a través de cada subportadora, que asegurará que la cantidad relativa de dispersión en el tiempo, causado por retardos de multitrayectoria, se reducirá. Pero cuando hay un gran número de subportadoras, la dificultad de sincronización en el receptor aumentará considerablemente.
- **Tiempo de Guarda (intervalo CP) y Duración del Símbolo:** Se debe establecer una buena relación entre el intervalo de CP y la duración del símbolo, a fin de que todas las multitrayectorias se resuelvan y que la cantidad de energía perdida debido al CP no sea demasiado alta. Como regla, el intervalo CP debe ser de dos a cuatro veces más grande que el retardo de propagación (RMS, *Root-Mean-Square*). La duración del tiempo del símbolo debe ser mucho mayor que el tiempo de guarda para minimizar la pérdida en la relación señal a ruido (SNR), aunque en una proporción razonable. No puede ser arbitrariamente grande, porque esto implica que se pueden agregar más subportadoras en el tiempo del símbolo, y más subportadoras aumentan la carga de procesamiento de señales, tanto en el transmisor como en el receptor, aumentando el coste y la complejidad del dispositivo resultante.
- **Espacio entre Subportadoras:** El espacio entre subportadoras debe mantenerse en un mismo nivel, a fin de lograr la sincronización. Este parámetro dependerá en gran medida del ancho de banda disponible y el número requerido de subcanales.
- **Tipo de Modulación por Subportadora:** Los requerimientos de desempeño pueden definir el esquema de modulación. La modulación adaptativa puede usarse para cubrir los requerimientos de desempeño en ambientes cambiantes.
- **Codificación FEC:** La elección del código FEC desempeña un papel fundamental. Una adecuada codificación FEC se asegurará de que el canal es resistente a los errores aleatorios.

CAPÍTULO IV

DESCRIPCIÓN DE LOS MODELOS

4.1 INTRODUCCIÓN

En el presente capítulo se definen los modelos de simulación empleados en este proyecto, cuyo principal objetivo es la investigación, para evaluar los resultados de la capa física IEEE 802.16 OFDM. Esto implica el modelado de la capa física, así como el ambiente de propagación. Los modelos de simulación elegidos para esta investigación son:

- **WiMAX 802.16e Physical Layer:** Modelo de construcción de la capa física de WiMAX, implementado utilizando *Matlab Simulink*. Basado en los estándares del IEEE y ETSI, la versión obtenida en Internet de este modelo, ver referencia [3], simula un canal downlink para el esquema de modulación QPSK.

En un proyecto anterior este modelo se modificó para simular, también, un canal Uplink y modulación 16QAM y 64QAM.

Este modelo, que en adelante será citado como "*Modelo sencillo*", utiliza los bloques ya definidos en por la herramienta *Matlab Simulink*, para implementar una interfaz de capa física basada en OFDM, con un transmisor, un receptor y, a diferencia del segundo modelo incluido en este trabajo, sólo un modelo de canal AWGN.

- **WiMAX/IEEE 802.16 OFDM Physical Layer:** Modelo que forma parte de la tesis "*Performance Evaluation of WiMAX/IEEE 802.16 OFDM Physical Layer*" [4], programado en *Matlab 6.0*. Este modelo, al igual que el modelo sencillo, implementa una interfaz *WirelessMAN-OFDM*, pero este modelo soporta los seis modelos de canales SUI (*Stanford University Interim*) y el canal AWGN, incluido para este trabajo.

Programado en módulos independientes, este modelo permite observar y manipular ampliamente las etapas y procesos de la simulación, y no solo los parámetros como en el caso del modelo sencillo.

En adelante este modelo será citado como modelo compuesto.

4.2 PARÁMETROS DEL SÍMBOLO OFDM

Hay dos tipos de parámetros OFDM (primarios y derivados) que definen completamente el símbolo OFDM. Los segundos pueden derivarse de los primeros, dada la dependencia que existe entre ellos. Para el caso de la capa física implementada en *Matlab*, los parámetros primarios son referidos como parámetros OFDM (*OFDM_params*), y son definidos al inicio de la simulación, y los parámetros derivados son calculados como parámetros *IEEE 802.16* (*IEEE 802.16_params*), a los cuales puede accederse de forma global.

Los parámetros OFDM están definidos de la siguiente manera:

PARAMETRO OFDM	VALOR
Ancho de banda nominal del canal, BW	5.50,5,3.50,3,2.50,1.75,1.5,1.25 MHz.
Número de subportadoras utilizadas, N_{used}	200
Factor de muestro, n	8/7
Razón entre el tiempo del Guarda(CP), y el tiempo útil del símbolo, G	¼, 1/8, 1/16, 1/32

TABLA 4.1: PARÁMETROS OFDM O PRIMARIOS.

Y los parámetros *IEEE 802.16* son calculados con las siguientes fórmulas.

PARAMETRO <i>IEEE 802.16</i>	VALOR
N_{FFT} (la potencia de 2 más pequeña, pero mayor a N_{used})	255
Frecuencia de Muestreo, F_s	Floor(n.BW/8000) X 8000
Espaciado entre Subportadoras. Δf	F _s / N _{FFT}
Tiempo útil del símbolo, T_b	1/Δf
Tiempo del prefijo cíclico, T_g	G.T _b
Tiempo del símbolo OFDM, T_s	T _b +T _g
Tiempo de muestreo, T_{sampling}	T _b /N _{FFT}

TABLA 4.2: PARÁMETROS *IEEE 802.16* O DERIVADOS.

Para el modelo sencillo [3] los parámetros se modifican directamente sobre los bloques y sobre el archivo **data.m**

4.3 CONFIGURACIÓN DE LA CAPA FÍSICA

La figura 4.1 muestra el diagrama a bloques del modelo sencillo que se empleó en este proyecto. Compuesto por transmisor y receptor, este modelo se subdivide en una parte que procesa los datos binarios (gris claro), bloques que emplean la tecnología OFDM (blanco), y los bloques de normalización (gris oscuro); los bloques al centro del diagrama son para el cálculo, comparación y visualización de los datos transmitidos y recibidos.

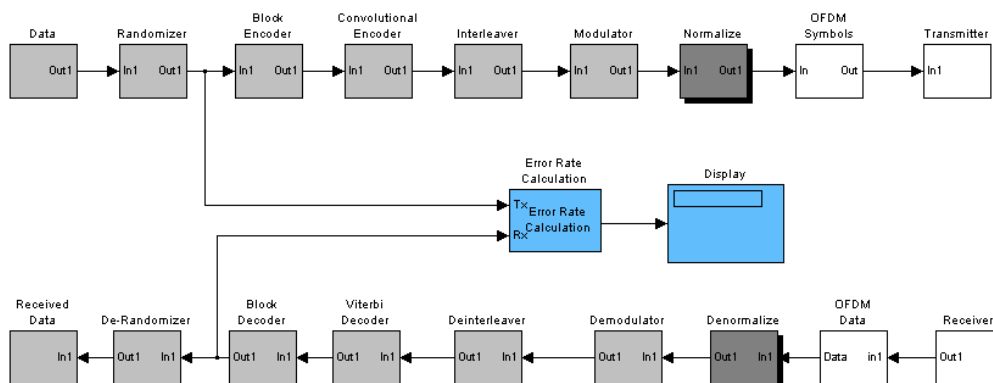


FIGURA 4.1: DIAGRAMA A BLOQUES DEL MODELO *SENCILLO* [3].

El modelo implementado en *Matlab* simula la capa física de la interfaz aérea WirelessMAN-OFDM del estándar IEEE 802.16-2004, implementando solamente las características básicas de la especificación. El diagrama a bloques de la figura 4.2 representa estas etapas.

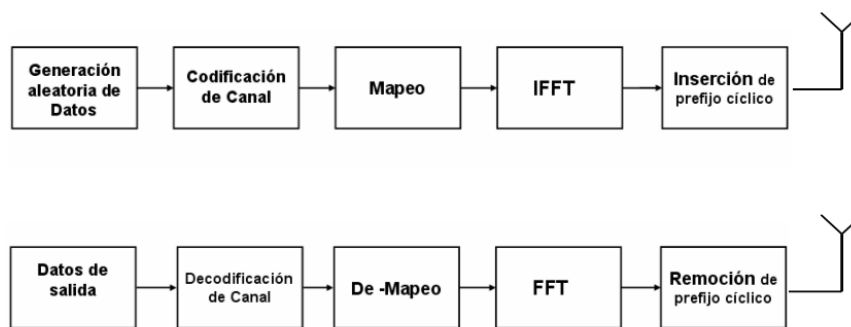


FIGURA 4.2: DIAGRAMA A BLOQUES DEL MODELO *COMPUESTO*.

La parte de codificación del canal está compuesta por tres etapas: *randomizer*, *FEC* e *interleaver*. El FEC se realiza en dos fases, mediante un código Reed-Solomon externo y el código convolucional interno. En la figura 4.3 se muestra el diagrama a bloque de la codificación de canal completa, codificación y decodificación,

En las subsecuentes secciones se describe brevemente cada bloque de la configuración de los modelos. Dado que la interfaz aérea WirelessMAN-OFDM ya fue descrita en el capítulo anterior, sólo se ampliará un poco dicha información desde el punto de vista de la implementación.

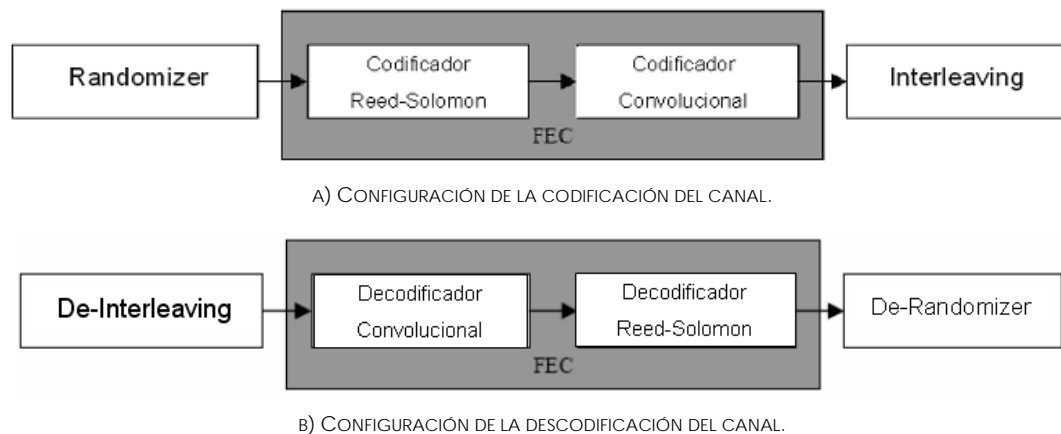


FIGURA 4.3: DIAGRAMA A BLOQUES DE LA CODIFICACIÓN DEL CANAL.

4.3.1 Randomizer

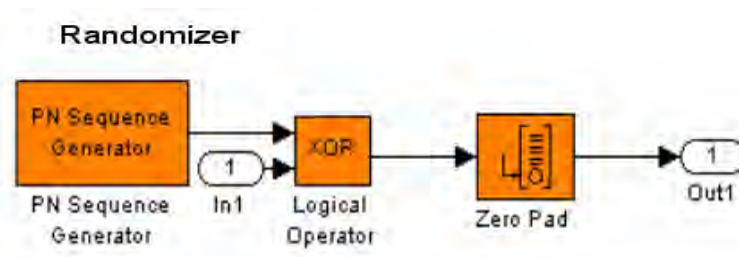
El *randomizer* o aleatorización de los datos se describe en la sección 3.3.1.1 de este proyecto.

Para el modelo compuesto se implementa un PRBS de 15 etapas con el polinomio generador $1+x^{14} + x^{15}$, y compuertas XOR en una configuración de retroalimentación, ver la figura 3.9 en la sección 3.3.1.1.

En el modelo sencillo esto se lleva a cabo en el bloque de "*Randomizer*", ver figura 4.4.

El "*PN Sequence Generator*", genera una secuencia lineal con retroalimentación con un polinomio generador como parámetro. En este caso el polinomio generador es el mismo que se definió para el modelo compuesto,

El "*XOR Logical Operator*" realiza la función booleana $A'B+AB'$.

FIGURA 4.4: BLOQUE *RANDOMIZER* DEL MODELO SENCILLO [3].

En el bloque "*Zero Pad*" se especifica el número de bits que se deben tener a la salida, para ello se agregan ceros hasta tener el número de bits deseado. Este valor cambia de acuerdo al tipo de modulación empleada:

- QPSK = 36 bytes * 8 bits = 288
- 16QAM = 72 bytes * 8 bits = 576
- 64QAM = 108 bytes * 8 bits = 864

En el de modulador se lleva a cabo el proceso inverso, Ver figura 4.5.

De-Randomizer

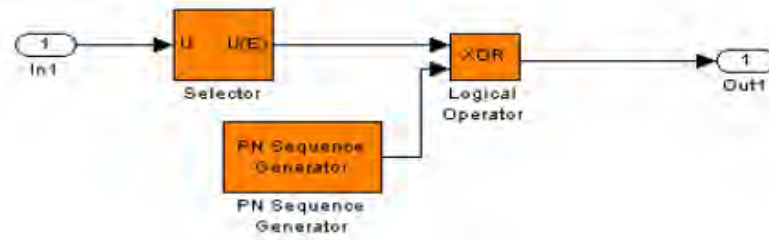


FIGURA 4.5: BLOQUE DE-RANDOMIZER DEL MODELO SENCILLO [3].

4.3.2 FEC

En la sección 3.3.1.2 de este proyecto se da una definición general del bloque FEC. Por lo que esta sección se enfocará en la implementación de los codificadores Reed-Solomon y Convolutacional

4.3.2.1 Codificador Reed-Solomon

Para el modelo compuesto, el codificador RS implementado es un código derivado de la codificación original RS(255, 239, t=8) sobre un GF(2⁸).

Los siguientes polinomios son usados como el generador de código y el generador de campo respectivamente:

$$G(x) = (x + \lambda^0)(x + \lambda^1) \dots (x + \lambda^{2^t-1}), \lambda = 02_{HEX} \tag{4.1}$$

$$P(x) = x^8 + x^4 + x^3 + x^2 + x + 1 \tag{4.2}$$

El codificador soporta código acortado y perforado (*punctured*) para facilitar un tamaño de bloque variable y una capacidad de corrección de errores también variable. Un bloque acortado de k bytes se obtiene agregando 239 - k bytes de ceros antes del bloque de datos y después de la codificación, estos 239 - k bytes de ceros se descartan.

En el modelo sencillo, el codificador Reed-Solomon se encuentra en el "Block Encoder"

En el "Bit to Integer Converter" se define el valor del número de bits que serán mapeados por cada entero. En este caso el valor correspondiente es 8 bits por cada byte. La figura 4.6 presenta la estructura del Block Encoder

Block Encoder

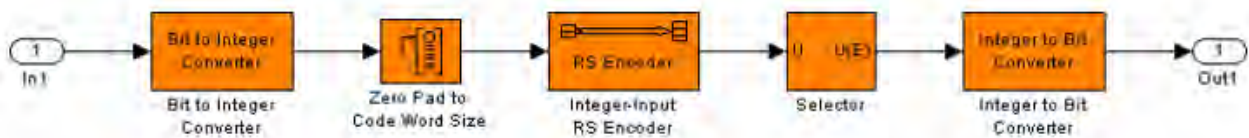


FIGURA 4.6: BLOQUE "BLOCK ENCODER" DEL MODELO SENCILLO [3].

En el bloque "Zero Pad to Code Word Size" se especifica el número de datos que se deben tener a la entrada, agregando bits "0" hasta tener el valor deseado. El estándar IEEE 802.16 especifica que dicha salida debe ser igual a 239, para este caso.

El bloque "Integer-Input RS Encoder" define los parámetros de arranque del codificador RS, así como también los polinomios generadores, que son los mismos que se definieron para el modelo compuesto.

En el convertidor de entero a bit se invierte el proceso llevado a cabo por el primer bloque.

4.3.2.1.1 Descodificación

Al igual que la mayoría de las funciones del lado del receptor, la decodificación para Reed-Solomon es el proceso inverso de la codificación, con una diferencia que podemos apreciar más fácilmente en el diagrama del bloque de decodificación, mostrado en la figura 4.7.

Existen dos bloques de "Zero Pad to Code Word", esto es porque la cadena que se recibe, para este caso, es de 255 y no de 239.

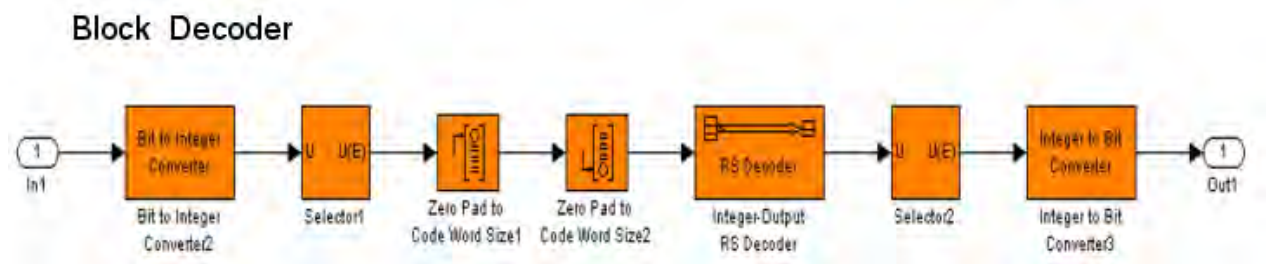


FIGURA 4.7: BLOQUE "BLOCK DECODER" DEL MODELO SENCILLO [3].

4.3.2.2 Codificación Convolutiva

En la sección 3.3.1.2.1 se describe el funcionamiento de la codificación convolutiva. Para el modelo compuesto esta codificación se lleva a cabo tal cual se describe en dicha sección.

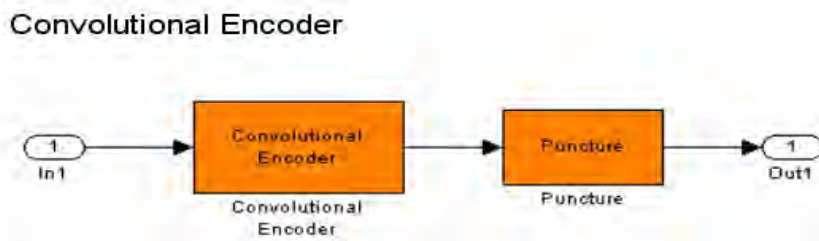


FIGURA 4.8: BLOQUE "CONVOLUTIONAL ENCODER" DEL MODELO SENCILLO [3].

El punteo o *puncturing* es el proceso mediante el cual se remueven algunos de los bits de paridad después de codificar con un código de corrección de errores. Esto ofrece una mayor tasa de corrección de errores, y una menor redundancia. Sin embargo, con el *puncturing* el mismo decodificador puede ser utilizado sin importar cuantos bits hayan sido eliminados, así, este proceso aumenta considerablemente la flexibilidad del sistema sin incrementar significativamente su complejidad. En la tabla 3.5 de la sección 3.3.1.2.1 se presenta los patrones de punteo para el código CC.

4.3.2.1 Decodificación

Existen dos algoritmos básicos para la descodificación de códigos convolucionales: el algoritmo de Viterbi y el algoritmo de decodificación secuencial. Éste último ofrece una cierta ventaja cuando los códigos son profundos, es decir donde K es muy grande, aunque el proceso de decodificación tiene una duración variable. En el algoritmo de Viterbi se emplea un tiempo de decodificación fijo, lo que es una ventaja si se quiere implementar mediante hardware. Pero este algoritmo, entre más profundo sea el código convolucional, requerirá más recursos computacionales para su implementación, ya que sus requerimientos crecen exponencialmente, por lo que en la práctica sólo se implementa para códigos convolucional con $K \leq 9$.

El algoritmo de decodificación de Viterbi está basado en los diagramas de Trellis. El **diagrama de Trellis** es un diagrama en forma de red, donde cada línea horizontal corresponde con uno de los estados del codificador y cada línea vertical correspondería a uno de los niveles del árbol del código. En la figura 4.9 se muestra un ejemplo de un diagrama de Trellis.

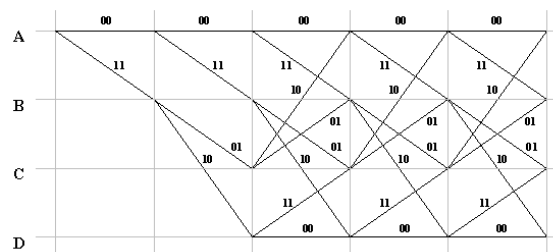


FIGURA 4.9: EJEMPLO DE DIAGRAMA DE TRELLIS PARA UN CODIFICADOR (2,1,3) [w4].

Este algoritmo intenta reconstruir las transiciones por las que pasó la información al ser del tal forma que se pueda recuperar la información original. En la figura 4.10 se presenta el bloque de "Viterbi Decoder."

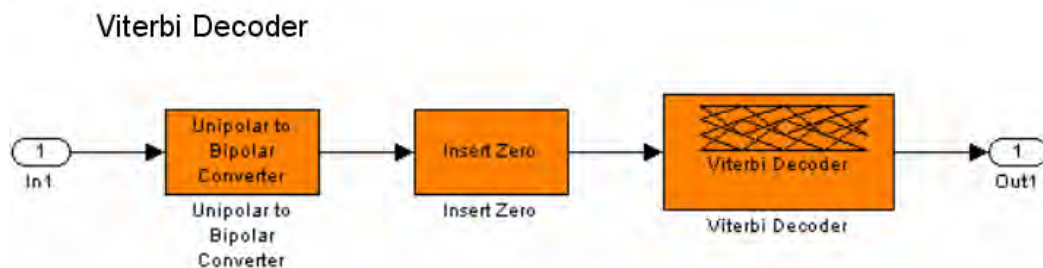


FIGURA 4.10: BLOQUE "VITERBI DECODER" DEL MODELO SENCILLO [3].

4.3.2.3 Interleaver

En este modulo básicamente se reordenan los bits de manera que los bits de una misma palabra no queden juntos. Esto con el fin de evitar que las ráfagas de errores, comunes en los ambientes inalámbricos, dañen por completo la información o que sea imposible recuperarla sin retransmisión.

La implementación en *Matlab* del *interleaver* fue realizada calculando el índice del valor de los bits después de la primera y segunda permutación usando respectivamente las siguientes ecuaciones:

$$fk = (Ncbps/12).kmod12+floor(k/2) \quad k = 0,1,2,\dots, Ncbps1 \quad (4.3)$$

$$s_k = s \cdot \text{floor}(fk/s) + (mk + N_{cbps} - \text{floor}(12 \cdot mk/N_{cbps})) \bmod(s) \quad k=0,1,2,\dots, N_{cbps}-1 \quad (4.4)$$

donde $s = \text{ceil}(N_{cpc}/2)$, mientras que N_{cpc} es el número de bits codificados por subportadora i.e. 1,2,4 o 6 para BPSK, QPSK, 16QAM o 64QAM respectivamente.

Para esta implementación el número predeterminado de canales es 16.

El receptor también realiza la operación inversa tras la permutación de dos pasos utilizando respectivamente las ecuaciones siguientes:

$$f_j = s \cdot \text{floor}(j/s) + (j + \text{floor}(12 \cdot j/N_{cbps})) \bmod(s) \quad j=0,1,\dots, N_{cbps}-1 \quad (4.5)$$

$$s_j = 12 \cdot f_j - (N_{cbps} - 1) \cdot \text{floor}(12 \cdot f_j/N_{cbps}) \quad j=0,1,2,\dots, N_{cbps}-1 \quad (4.6)$$

En la figura 4.11 se muestra el bloque de *Interleaver*, y en la figura 4.12 el de *De-Interleaver*.

Interleaver



FIGURA 4.11: BLOQUE "INTERLEAVER" PARA EL MODELO SENCILLO [3].

De- Interleaver



FIGURA 4.12: BLOQUE "DEINTERLEAVER" PARA EL MODELO SENCILLO [3].

4.3.3 Modulación

4.3.3.1 Constellation Mapper o Mapeo de Constelaciones

Después que los bits de datos son intercalados entran de forma serial pasan al mapeo de constelaciones. El modelo compuesto soporta mapeo de constelaciones para BPSK y mapeo grey para QPSK, 16QAM y 64QAM. Los puntos de constelación complejos son normalizados con el factor de multiplicación específico de cada esquema de modulación, como por ejemplo 2 para QPSK, tal que una potencia promedio igual es calculada para los símbolos.

4.3.3.2 IFFT

Después del mapeo grey los datos son enviados al bloque IFFT para un mapeo en el dominio del tiempo. El mapeo en el dominio del tiempo requiere la función de transformada rápida de Fourier Inversa (IFFT). El modelo compuesto incorporó una función "ifft" para hacerlo. Este bloque ofrece un vector de 256 elementos, donde cada número complejo representa un símbolo OFDM. Por último, y sólo después de bloque IFFT, se agrega el prefijo cíclico, debido a que, como se menciona en el inciso 3.4.2, es una copia espejo del símbolo OFDM en el dominio del tiempo.

En las figuras 4.13 y 4.14 se muestra donde pasan los datos por IFFT en el transmisor y por FFT en el receptor respectivamente.

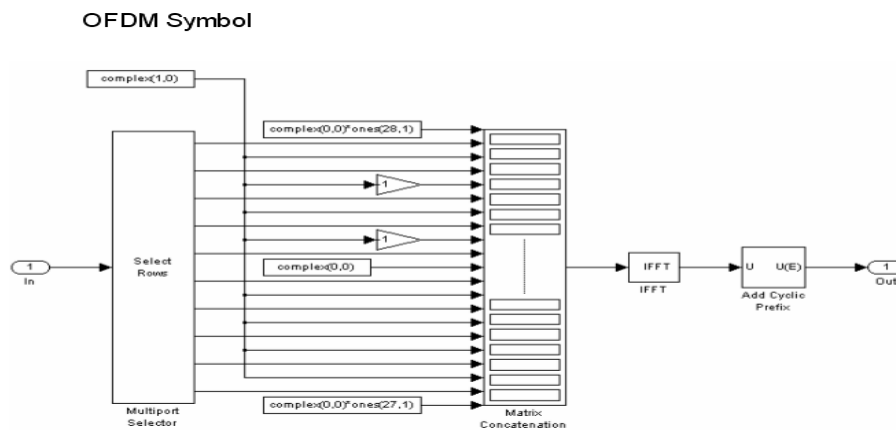


FIGURA 4.13: BLOQUE "OFDM SYMBOL" PARA EL MODELO SENCILLO [3].

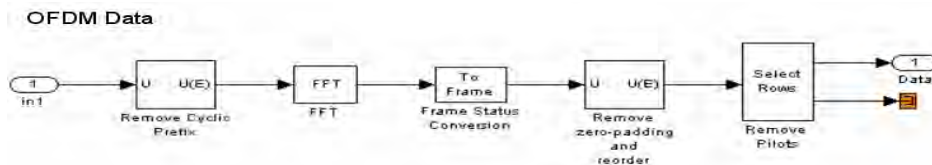


FIGURA 4.14: BLOQUE "OFDM DATA" PARA EL MODELO SENCILLO [3].

En el lado del receptor, para ambos modelos, el proceso es el inverso. Primero se remueve el prefijo cíclico antes de volver la señal al dominio de la frecuencia, a través de la FFT, ya que el prefijo se define en el dominio de tiempo. Después de FFT, se remueve el relleno y se separan los pilotos del resto de la información.

4.4 MODELADO DEL CANAL.

Una descripción precisa del canal inalámbrico es un requisito esencial en la evaluación del desempeño de un sistema. Cualquier modelo de canal depende, en gran medida, de la arquitectura del radio. Los canales inalámbricos se caracterizan por:

- Pérdidas por trayectoria (*Path Loss*).
- Shadowing (*log-normal shadowing*).
- Retardo de propagación por multitrayectoria (*Multipath Delay Spread*).
- Características de desvanecimiento (*Fading*).
- Efecto *Doppler* (*Doppler Spread*).
- Interferencia de canal adyacente y co-canal.

Para todos los canales las deficiencias son aleatorias, y sólo es posible definir estas características de forma estadística, es decir, en términos de la media y varianza. Todos los parámetros del canal dependen de diferentes factores, como el terreno, la topología, la densidad de la vegetación, la altura de la antena, la velocidad del viento, entre muchos otros factores.

4.4.1 *Path Loss*

El *path loss* depende en gran medida del tipo de terreno y la densidad de los obstáculos.

El modelo Hata-Okumura de *path loss* es uno de los más utilizados para predecir la fuerza de la señal en simulaciones. Este modelo es válido en la gama de frecuencias de 500 a 1500 MHz; el receptor puede estar a distancias superiores a 1 km de la BS, y la altura para las antenas de las estaciones base es superior a 30 metros. El *path loss* se puede dividir en categorías por regiones, que dependen de los factores antes mencionados. Estas categorías se basan en el tipo terreno y los obstáculos en el camino de la propagación. En la actualidad se consideran tres tipos de terreno:

Tipo A: Colinas pequeñas con moderada o alta densidad de obstáculos (casas, edificios árboles), como ciudades densamente pobladas, zonas montañosas o con grandes bosques. Este tipo de terreno se asocia con el mayor *path loss*.

Tipo b: Colinas grandes con baja densidad de obstáculos, o plano con moderada o alta densidad de obstáculos, como ciudades pequeñas.

Tipo c: Plano con baja densidad de obstáculos, como algunas comunidades rurales. Este tipo de terreno se asocia con el menor *path loss*.

La media de *path loss* (PL en decibeles), para distancias cortas está dada por:

$$PL = A + 10\gamma \log_{10} (d/d_0) + s \quad d > d_0 \quad (4.7)$$

donde:

$$A = 20 \log_{10} (4 \pi d_0 / \lambda) \quad (4.8)$$

con λ dada en metros; el exponente de *path loss*, γ , dado por:

$$\gamma = (a - b * h_b + c / h_b) \quad (4.9)$$

con la altura de la antena para la estación base, H_b , $10\text{m} < hb < 80\text{m}$, y a , b y c son constantes dependientes del tipo de terreno; d_0 es la distancia de referencia escogida; s representa el efecto del ensombrecimiento o *shadowing*, para el cual se asume una distribución normal con desviación estándar entre 8 y 10 dB.

PARÁMETRO	TERRENO		
	TIPO A	TIPO B	TIPO C
a	4.6	4.0	3.6
$b(\text{m}^{-1})$	0.0075	0.0065	0.005
c (m)	12.6	17.1	20

TABLA 4.3: CONSTANTES DEPENDIENTES DEL TERRENO.

4.4.2 Retardo de Propagación por multitrayectoria (*Multipath Delay Spread*)

Debido a la natural propagación sin línea de vista (NLOS) de WirelessMAN OFDM, el canal de modelo de *Matlab* presenta muchos retrasos por efecto de multidirección. Esto es el resultado de la dispersión natural en el ambiente. El retardo de propagación es un parámetro utilizado para denotar el efecto de propagación multitrayectoria. Depende del terreno, distancia, la direccionalidad de la antena y otros factores. El valor del retardo RMS varía desde unos 10 ns a milisegundos.

4.4.3 Características de desvanecimiento (*Fading*).

El desvanecimiento o *fading*, se refiere a la variabilidad en el nivel de la señal, que puede ser definida por algunos modelos de distribución estadística. En un entorno de propagación multitrayectoria, la señal recibida experimenta fluctuaciones en su amplitud, fase y el ángulo de llegada. El efecto es descrito por el término *fading*. En el modelo compuesto las antenas, tanto la del receptor como la de el transmisor, están fijas el modelo del canal sólo presenta un *fading* de "baja escala". El *fading* de baja escala se refiere a los cambios en la amplitud y fase de la señal, que pueden ser el resultado de pequeños cambios (tan pequeño como media longitud de onda), en el espacio entre un receptor y un transmisor.

Cuando el desvanecimiento es provocado por el rebote de la señal al recorrer varios caminos, de la estación base al usuario móvil, y estos ecos predominan sobre la señal en trayectoria directa, se le conoce como "Desvanecimiento de *Rayleigh*". Y cuando la señal en trayectoria directa es más fuerte que los ecos, entonces se conoce como "Desvanecimiento de *Rice*".

Para el modelo del canal en *Matlab*, se considera un desvanecimiento de *Rice*. El parámetro clave de esta distribución es el factor k , que se define como la relación directa entre el componente de energía y el componente de dispersión de energía. En la tabla 4.7, que se encuentra en la sección 4.4.5, se listan los valores del factor k para una antena omnidireccional.

4.4.4 Efecto *Doppler*

El efecto *doppler* consiste en una variación en la frecuencia de la portadora que ocurre debido al movimiento relativo entre el transmisor y el receptor, es decir, cuando el transmisor o el receptor están en movimiento. Conforme se acercan el receptor percibe un aumento en la frecuencia de la señal que se transmite, y si se alejan el receptor percibe una atenuación de la señal. En una red fija el efecto *doppler* se centra en el movimiento de los objetos en el entorno, mesas, sillas, personas, entre otros.

La siguiente ecuación permite calcular el efecto *doppler*.

$$\Delta f = f c (v/c) \quad (4.10)$$

Donde:

Δf : Variación de la frecuencia.

v : Velocidad relativa entre el transmisor y el receptor.

c : Velocidad de la luz.

4.4.5 Modelos de Canales SUI (*Stanford University Interim*)

Es evidente que hay muchas combinaciones posibles de parámetros para el modelado de un canal. Por ello se seleccionó un conjunto de seis canales, conocidos como modelos de canales SUI, para los tres tipos de terreno que son típicos en EE.UU [16]. Los canales SUI pueden utilizarse para el diseño, desarrollo y ensayo de tecnologías fijas inalámbricas de banda ancha. El modelo compuesto implementa estos seis perfiles y un canal AWGN, mientras que el modelo sencillo solamente utiliza un canal AWGN.

Los parámetros para estos perfiles fueron seleccionados sobre la base de los modelos estadísticos descritos. En las tablas 4.3 y 4.4 se muestran algunos de los parámetros y las características de los seis canales SUI.

TIPO DE TERRENO	CANAL SUI
C	SUI 1, SUI2
B	SUI 3, SUI4
A	SUI 5, SUI6

TABLA 4.4 : CANAL SUI POR TIPO DE TERRENO.

Doppler	Retardo de Propagación bajo	Retardo de Propagación Moderado	Retardo de Propagación Alto
Bajo	SUI1, SUI 2, SUI 3		SUI5
Alto		SUI4	SUI6

TABLA 4.5: CARACTERÍSTICAS GENERALES DE LOS CANALES SUI.

La estructura genérica de los modelos de canal SUI puede observarse en la figura 4.15.

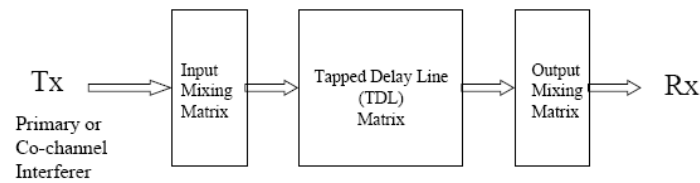


FIGURA 4.15: ESTRUCTURA GENÉRICA DE LOS MODELOS DE CANAL SUI [16].

Apoyándose en esta estructura general y definiendo un escenario es posible simular los seis canales SUI, que son representativos de los canales reales.

Si se asume un escenario con las siguientes características:

- Tamaño de célula: 7Km.
- Altura de la antena de BS: 30 Mts.
- Altura de la antena de recepción: 6m.
- *Beamwidth* de la antena de BS: 120°.
- *Beamwidth* de la antena de recepción: omnidireccional.
- Polarización: sólo Vertical.
- 90% la cobertura celular con 99,9% de fiabilidad en cada lugar cubierto.

Con este escenario y utilizando el modelo de cada canal, en las tablas 4.6 a 4.8 se listan los valores de los parámetros, obtenidos de forma estadística, definidos para cada uno de los seis canales SUI.

CANAL	TAP1	TAP 2 (μ s)	TAP 3 (μ s)	RETARDO DE PROPAGACION RMS (ms)
SUI1	0	0.4	9.9	0.111
SUI2	0	0.4	11	0.202
SUI3	0	0.4	0.9	0.264
SUI4	0	1.5	4	1.257
SUI5	0	4	10	2.842
SUI6	0	14	20	5.240

TABLA 4.6: RETARDO DE PROPAGACIÓN PARA LOS MODELOS DE CABAL SUI.

CANAL	TAP 1	TAP 2 (dB)	TAP 3 (dB)
SUI1	0	-15	-20
SUI 2	0	-12	-15
SUI 3	0	-5	-10
SUI 4	0	-4	-8
SUI 5	0	-5	-10
SUI 6	0	-10	-14

TABLA 4.7: TAP POWER (ANTENA OMNI DIRECCIONAL), DE CANALES SUI

CANAL	TAP 1	TAP 2	TAP 3
SUI1	4	0	0
SUI 2	2	0	0
SUI 3	1	0	1
SUI 4	0	0	0
SUI 5	0	0	0
SUI 6	0	0	0

TABLA 4.8: 90 % DE FACTOR K (ANTENA OMNI DIRECCIONAL), DE CANALES SUI

4.4.6 Implementación de los modelos de canal SUI

El objetivo de la implementación del modelo es simular los coeficientes del canal. Los coeficientes del canal, con la distribución especificada y la densidad de la potencia espectral, se generan utilizando el método de filtrado de ruido.

Un grupo de números con una distribución gaussiana compleja y una media de cero, es generado con una varianza de 0.5 para la parte real y la parte imaginaria por cada tap (punto de conexión de red), para lograr la potencia media total de esta distribución, que es 1. Con esta idea se toma una distribución *Rayleigh* (equivalente a una *Rice* con $K = 0$) para la magnitud de los coeficientes complejos. En el caso de una distribución *Rice* ($K > 0$) un componente m de trayectoria constante se agrega al conjunto de coeficientes *Rayleigh*. El factor K define el ratio de potencias entre la parte constante y la parte variable. La distribución de potencia se muestra a continuación:

La potencia total p para cada tap:

$$p = |m|^2 + \sigma^2 \quad (4.11)$$

donde: m es la constante compleja y σ^2 es la varianza de la serie gaussiana compleja, la relación de la potencia es:

$$k = \frac{|m|^2}{\sigma^2} \quad (4.12)$$

De las dos ecuaciones anteriores, se obtiene la potencia de la serie gaussiana compleja:

$$\sigma^2 = p \frac{1}{k+1} \quad (4.13)$$

y la potencia de la parte constante es:

$$|m|^2 = p \frac{k}{k+1} \quad (4.14)$$

El modelo de canal SUI define una función de densidad de potencia espectral (PSD, *Power Spectral Density*), para el componente de dispersión de los coeficientes del canal que está dada por:

$$S(f) = \begin{cases} 1 - 1.72f_0^2 + 0.785f_0^4 & |f_0| \leq 1 \\ 0 & |f_0| > 1 \end{cases} \quad (4.13)$$

Donde, la función es parametrizada por una frecuencia *doppler* máxima f_m y

$$f_0 = f/f_m \quad (4.14)$$

Para generar un conjunto de coeficientes de canal con esta función PSD los coeficientes originales son correlacionados con un filtro cuya respuesta en amplitud es:

$$H(f) = S(f) \quad (4.15)$$

Para una implementación eficiente, un filtro no recursivo y un método de adición superpuesta en dominio de la frecuencia es usado.

No hay componentes de frecuencia superiores a f_m (para la formula de construcción $S(f)$); pero el canal puede ser representado con un mínimo de frecuencia de muestreo de $2f_m$, de acuerdo con el teorema de Nyquist. Por esta razón se escogió una frecuencia de muestre igual a $2f_m$.

La potencia del filtro tiene que ser normalizado a 1, a fin de que la potencia total de la señal de salida sea igual a la de entrada.

CAPÍTULO V

RESULTADOS DE LA SIMULACIÓN

5.1 INTRODUCCIÓN

En este capítulo, se presentan los resultados de las simulaciones, así como algunas de las conclusiones que se derivan de las mismas. El objetivo es analizar los resultados de ambos modelos, el modelo a bloques de *SIMULINK* y el modelo programado en *Matlab*, para obtener una comprensión más exacta de la operación de la capa física de una red WirelessMAN. A través de varias comparaciones entre los resultados de las simulaciones, obtenidos con diferentes parámetros de simulación, algunos comentarios sobre la utilidad de estos parámetros, y opciones que ofrecen una visión más amplia sobre cómo obtener un mayor rendimiento en la transmisión. Diferentes modelos de canal y escenarios se aplicaron en diferentes simulaciones, a fin de que el fenómeno de la atenuación pueda ser analizado.

Varias simulaciones, con una variedad de modulaciones (QPSK, 16QAM y 64QAM) y parámetros del canal (SUI1 a SUI5), se llevaron a cabo y, en cada ocasión, se grafican el BER y BLER.

Para la simulación con el modelo sencillo se realizaron cambios en el archivo "Data.m", para que el modelo soportara los esquemas de modulación de 16QAM y 64QAM, y directamente sobre los bloques. Estos cambios obedecen a que cada esquema de modulación requiere vectores y datos específicos, diferentes a los predeterminados en el modelo.

Un ejemplo de esto es el valor asignado a N_{cbps} que es distinto para cada modulación:

- para QPSK, $N_{cbps} = 384$,
- para 16QAM, $N_{cbps} = 768$ y
- para 64QAM $N_{cbps} = 1156$.

Otro ejemplo de esto es el vector de *puncture* que depende de la tasa de la codificación convolucional (CC), en la figura 5.1 se muestra la ventana de parámetros para el bloque de *Puncture*, y el vector definido corresponde a una tasa de CC de 5/6.

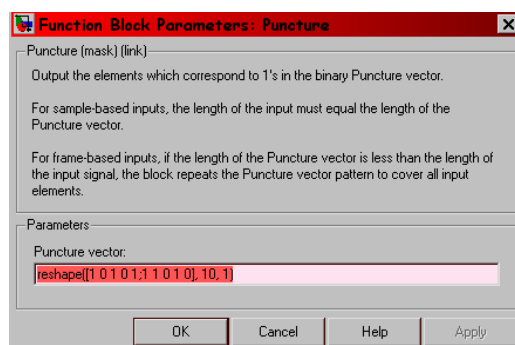


FIGURA 5.1: PARÁMETROS DE *PUNCTURE*, CODIFICACIÓN CC 5/6.

En el Apéndice A, se presentan las versiones del archivo 'Data.m' para QPSK, 16QAM y 64QAM. Los vectores de codificación, tanto convolucional como de Reed-Solomon, son definidos por el estándar y se muestran en la tabla 3.6 de la sección 3.3.1.2.2, y la tabla 5.1 de este capítulo.

El modelo compuesto fue diseñado en módulos. Cada una de las funciones que componen el transmisor, canal y receptor se encuentran en archivos '.m' por separado. El proceso principal liga todos estos módulos y provee los parámetros primarios para la modulación.

El proceso principal del modelo compuesto que se desarrollo para este trabajo, no estaba incluido en los textos de referencia, ofrece una relativa facilidad para la definición de los parámetros primarios. La potencia, el canal, la modulación, el tiempo de guarda y algunos parámetros más son definidos por el usuario en este proceso. También en este módulo se definen los datos de entrada, que son generados por un proceso aleatorio que entrega "n" palabras de 8 bits, donde n es un valor definido por el usuario.

5.2 LA SIMULACIÓN

En esta sección se presenta una comparación entre los diferentes esquemas de modulación y tasas de codificación utilizados en los simuladores.

Los resultados de la simulación para los diferentes esquemas de modulación y tasas de codificación, así como los distintos canales se muestran en esta sección. Tal como se especifica en el estándar 802.16, las redes basadas en este estándar permiten siete diferentes esquemas de modulación y tasas de codificación, que se resumen a continuación, Tabla 5.1.

MODULACIÓN	BLOQUE DE DATOS SIN CODIFICAR (BYTES)	BLOQUE DE DATOS, CODIFICADOS (BYTES)	TASA DE CODIFICACIÓN GLOBAL	CÓDIGO RS	TASA DE CODIFICACIÓN CC
BPSK	12	24	1/2	(12,12,0)	1/2
QPSK	24	48	1/2	(32,24,4)	2/3
	36	48	3/4	(40,36,2)	5/6
16QAM	38	96	1/2	(64,48,8)	2/3
	72	96	3/4	(80,72,4)	5/6
64QAM	96	144	2/3	(108,96,6)	3/4
	108	144	3/4	(120,108,6)	5/6

TABLA 5.1: ESQUEMAS DE MODULACIÓN Y TASAS DE CODIFICACIÓN SOPORTADOS POR EL ESTÁNDAR 802.16.

Debe tenerse en cuenta que al utilizar valores aleatorios los resultados no son únicos, es decir, éstos pueden variar entre una simulación y otra, aun cuando los escenarios y parámetros del sistema sean iguales. Por ello para cada caso se realizaron alrededor de 15 simulaciones. Esto es, por cada esquema de modulación y tasa de codificación, para cada valor de SNR.

5.2.1 Bit Error Rate (BER)

Para poder comparar cómo operaría cada uno de los escenarios y los resultados que se obtendrían en la práctica con los diferentes esquemas de modulación, se ha simulado el comportamiento de todos ellos bajo condiciones similares de operación analizando la tasa de bits erróneos, BER, para cada modelo bajo diferentes relaciones señal a ruido de operación.

El *bit error rate* es la tasa de bits en error, es decir el número de bits en error entre el total de bits recibidos, el cual es diferente para cada valor de SNR.

En esta sección se presentan las gráficas de BER vs. SNR, en el caso del modelo sencillo, un canal AWGN, en la figura 5.2 se pueden observar las curvas BER para seis de los siete diferentes esquemas de modulación y tasas de codificación definidos en el estándar 802.16.

Las figuras 5.6, 5.7, 5.8, 5.9 y 5.10 muestran la relación BER vs. SNR para los canales SUI 1, SUI 2, SUI 3, SUI 4 y SUI 5, simulados con el modelo compuesto, y los siguiente parámetros primarios:

- Datos de entrada $3,000 \times 8$ bits, esto es para todos los canales.
- Un ancho de banda nominal, $BW = 2.5\text{MHz}$ para todos los canales.
- Tasa de tiempo de guarda, $G = 1/16$ para SUI 1, SUI 2, SUI 3 y SUI 4, y $G = 1/8$ para SUI 5.

La figura 5.2, muestra las curvas BER para el modelo a bloques en *SIMULINK*, en el que se define un canal AWGN, cuyo diagrama a bloques se muestra en la figura 5.3.

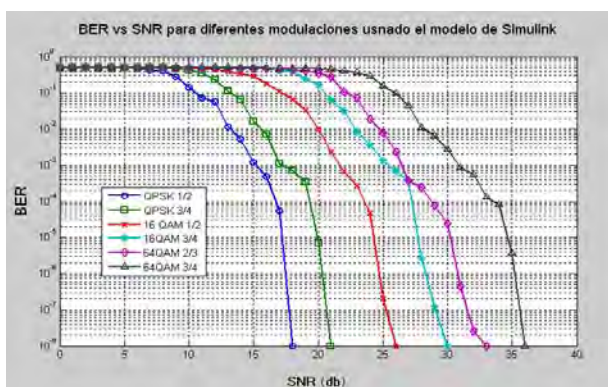


FIGURA 5.2: BER VS SNR USANDO EL MODELO SENCILLO

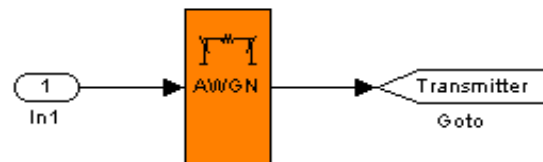


FIGURA 5.3: MODELO A BLOQUES DEL CANAL AWGN DEFINIDO EN EL MODELO SENCILLO.

La figura 5.4 muestra las curvas BER obtenidas para el canal AWGN definido en el modelo compuesto. Este canal se implemento mediante la función que el software define: *awgn(x,snr,'measured', dB)*. La información referente a la función "*awgn*" y los parámetros que ocupa se definen en la ayuda de *Matlab*, y los valores de los parámetros, tales como el valor de SNR o el número de datos, se obtienen mediante la simulación.

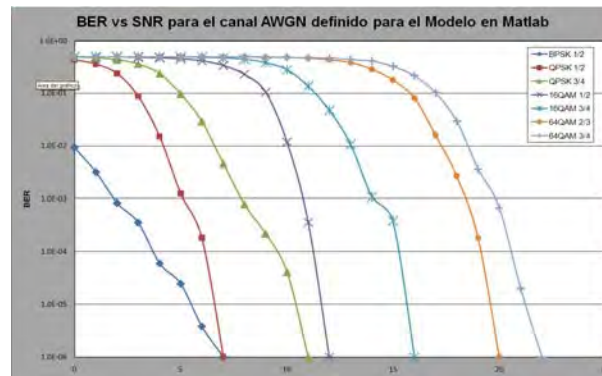


FIGURA 5.4: CURVAS BER PARA EL CANAL AWGN DEFINIDO EN EL MODELO COMPUESTO.

En el caso del modelo sencillo, figura 5.2, las curvas del BER tienden a cero más lentamente que las curvas para el canal AWGN del modelo compuesto, figura 5.4. En diversos artículos, incluidos en las referencias, para una red basada en el estándar 802.16d con un canal AWGN, el valor de BER es cero alrededor de un SNR = 25 dB, para todos los escenarios, mientras que en la gráfica de la figura 5.2, las curvas para las modulaciones 16QAM $\frac{3}{4}$, 64 QAM $\frac{2}{3}$ y 64 QAM $\frac{3}{4}$ superan este margen, y tienden a cero más lentamente que las curvas obtenidas para los canales SUI1y SUI2, figura 5.6 y figura 5.7 respectivamente. Las curvas BER mostradas en la gráfica de la figura 5.4 tienden a cero con más rapidez, y para todos los escenarios las curvas son cero alrededor de SNR=22 dB.

Para el canal AWGN del modelo compuesto, a diferencia de los canales SUI, el umbral donde el valor BER es cero no varía demasiado, a menos que entre una simulación y otra se modifique directamente la función "awgn".

En la siguiente gráfica, figura 5.5, se muestra cómo el tamaño de la información de entrada afecta al BER. Este tipo de gráfica se conoce como de estabilidad o estado estable, ya que permite observar el comportamiento del BER para diferentes tamaños de datos de entrada, en este caso en un rango desde 100 palabras de información (bytes), a 1,000,000 palabras de información, y con SNR = 7 dB.

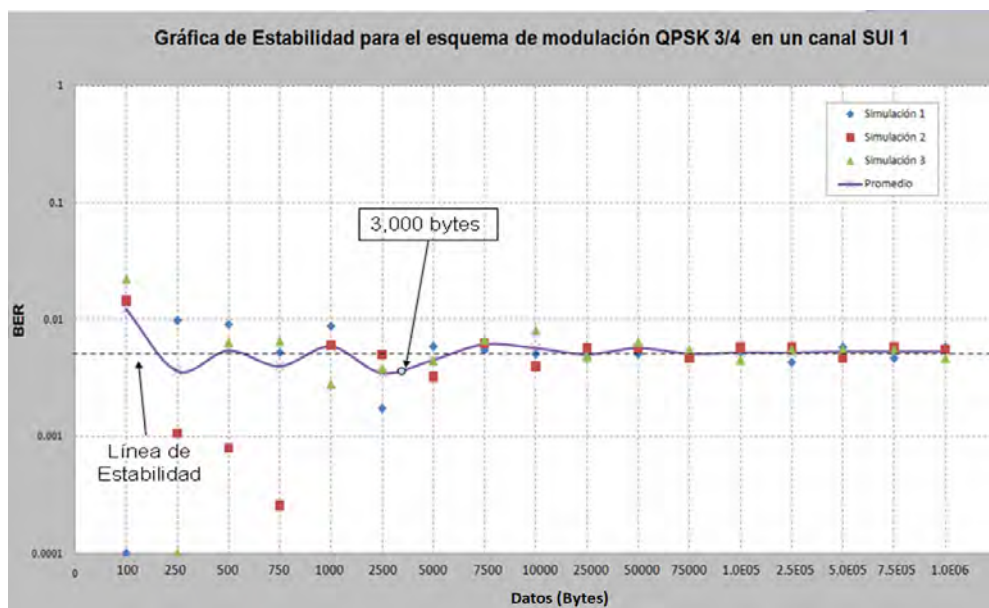


FIGURA 5.5: GRÁFICA DE ESTABILIDAD PARA QPSK $\frac{3}{4}$ USANDO EL MODELO COMPUESTO.

En la gráfica de la figura 5.5 podemos ver que, con pocos bytes de entrada, la diferencia entre los valores de cada simulación es grande. Conforme se aumentan los bytes de entrada la diferencia entre los valores se reduce. En teoría, existe un número de palabras de información para el cual la diferencia entre simulaciones es despreciable. En esta gráfica también se puede observar que la curva se va suavizando, es decir, que el valor del BER para dos distintos valores de entrada es muy similar, lo que implica que el sistema ha alcanzado el estado estable. El tiempo de simulación es directamente proporcional al tamaño de la entrada, cuando la entrada es de 100 bytes el tiempo de simulación, para cada uno de los valores BER, rara vez excede los 10 minutos, mientras que con 1,000,000 tarda más de una semana.

Las gráficas mostradas en las figuras 5.6 a 5.10, como se menciona al inicio de esta sección, son las curvas BER para los diferentes esquemas de modulación, para cinco de los seis modelos SUI y una entrada de 3,000 bytes. Con esta entrada de 3,000 bytes el sistema ya ofrece un nivel de estabilidad aceptable para los propósitos de esta tesis, como se observa en la figura 5.5, y el tiempo de la simulación es relativamente corto.

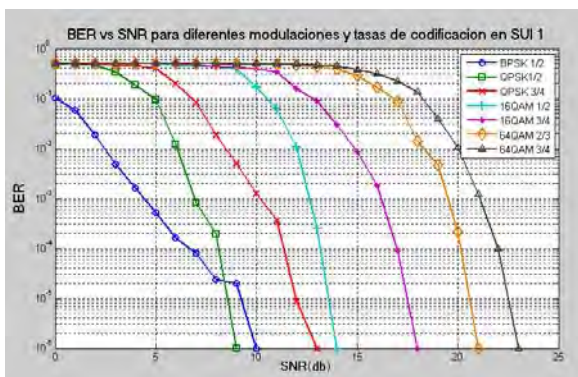


FIGURA 5.6: BER vs SNR EN UN CANAL SUI 1.

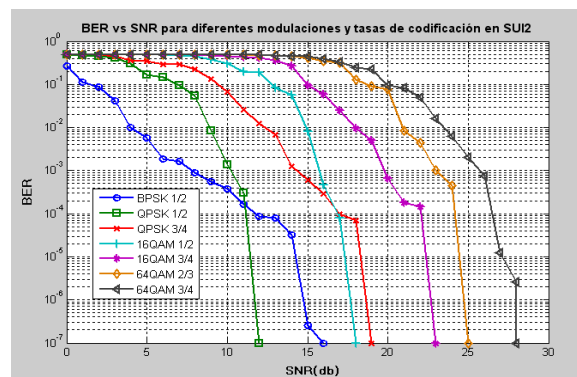


FIGURA 5.7: BER vs SNR EN UN CANAL SUI 2.

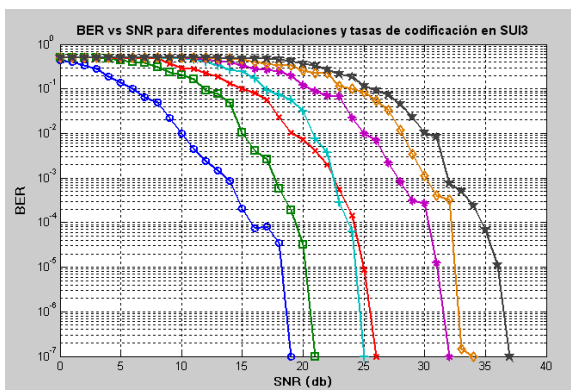


FIGURA 5.8: BER vs SNR EN UN CANAL SUI 3.

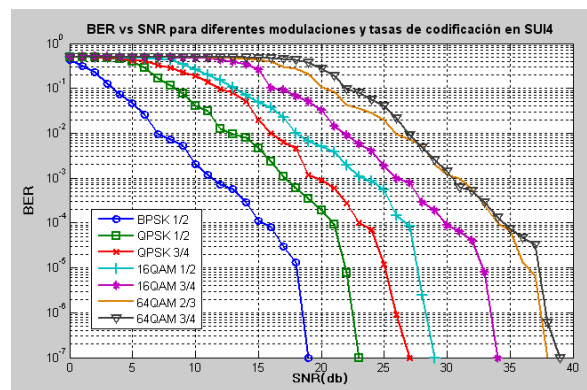


FIGURA 5.9: BER vs SNR EN UN CANAL SUI 4.

En las gráficas 5.8 y 5.9 es posible observar la similitud de las curvas BER para los canales SUI 4 y SUI 3, en donde la mayor diferencia se da en el punto donde el BER comienza a descender, en la parte alta de las curvas.

Siguiendo el orden de las gráficas anteriores para las distintos canales, se observa como las curvas del BER tiende a cero más lentamente; empezando con la gráfica del canal SUI 1, cuyas curvas BER son las mas cercanas al "ideal", hasta las curvas de los canales SUI4 y SUI 5 que, para este proyecto, son las más altas.

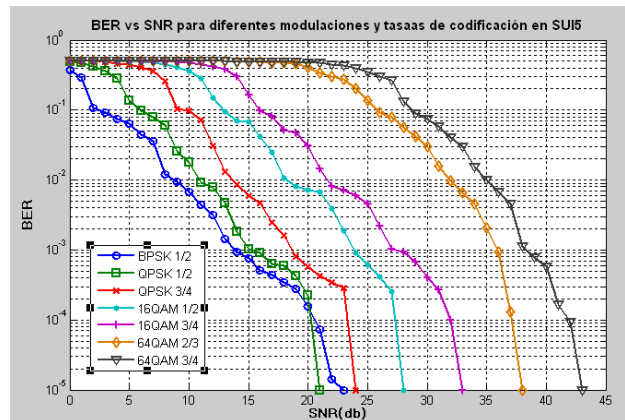


FIGURA 5.10: BER VS SNR EN UN CANAL SUI 5.

Estas curvas presentan diferentes comportamientos, como es mantenerse en un punto máximo, valores en el rango de 0.5, para varios valores de SNR, y caer drásticamente a la iteración siguiente, como por ejemplo las curvas en figura 5.6. Para los canales SUI 4 y SUI 5 empiezan a descender relativamente pronto, pero las curvas son menos pronunciadas y tardan más en llegar al cero. En la figura 5.11, se presentan las curvas BER para 64QAM 2/3, en los diferentes canales SUI, donde se observa más claramente las diferencias entre dichas curvas.

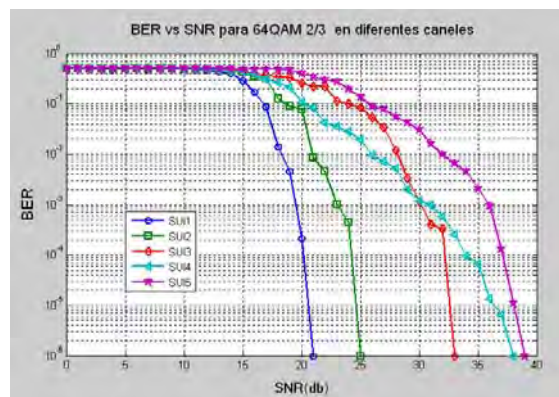


FIGURA 5.11: BER VS SNR PARA 64QAM 2/3 PARA DIFERENTES SUI.

Las curvas BER son importantes para ayudar a definir un valor de SNR donde el BER entra en un rango aceptable, alrededor de 10^{-3} . Al definir este umbral es posible mejorar la velocidad de transmisión de datos.

En la gráfica comparativa, figura 5.11, es posible observar la diferencia que existe entre los valores de SNR, para los distintos canales, donde BER alcanza el umbral de 10^{-3} . Para un canal SUI 1, en este escenario, esta valor es SNR = 19.5 dB aproximadamente, para el canal SUI 2, el umbral se logró cerca de SNR = 23 dB; y para SUI 5, que en comparación es el más alto, el umbral están en SNR = 37 dB.

La diferencia entre los canales SUI 3 y SUI 4 en esta gráfica es mínima, y pareciera que el umbral para este escenario es el mismo, SNR = 30, aunque anteriormente ya se había mencionado esta similitud.

5.2.2 Block Error Rate (BLER)

El *block error rate* es una relación entre el número de bloques erróneos y el número total de bloques recibido. El BLER puede ser definido en función del BER de la siguiente forma:

$$BLER = 1 - (1 - BER)^k \tag{5.1}$$

Donde k es el tamaño del bloque de datos sin codificar.

El BLER desempeña un papel muy importante en el análisis del desempeño de la capa física. Las figuras 5.12 a 5.16 muestran las curvas BLER para los diferentes esquemas de modulación y tasas de codificación mostrados en la tabla 5.1.

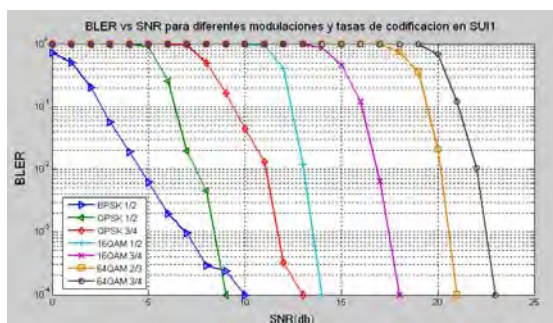


FIGURA 5.12: BLER vs SNR EN UN CANAL SUI 1.

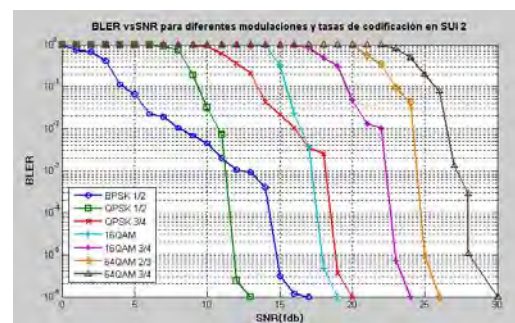


FIGURA 5.13: BLER vs SNR EN UN CANAL SUI 2.

Así como para BER se busca un valor para SNR donde el BER alcanza el umbral del rango 10^{-3} , para el BLER este umbral se ubica en el rango próximo a 10^{-2} .

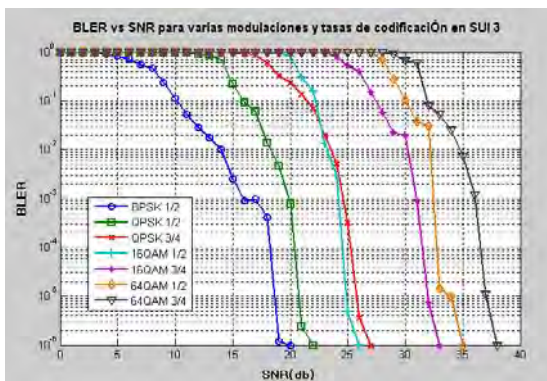


FIGURA 5.14: BLER vs SNR EN UN CANAL SUI 3.

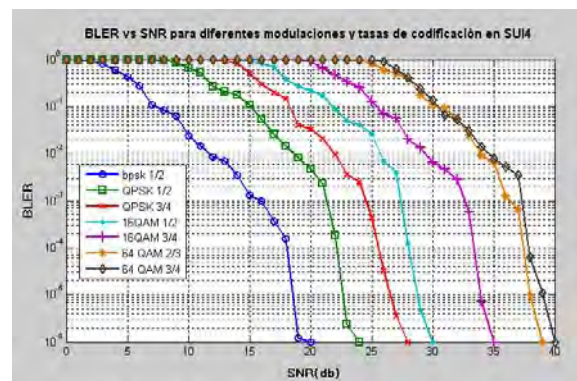


FIGURA 5.15: BLER vs SNR EN UN CANAL SUI 4.

Al igual que con el BER, las curvas BLER para SUI3 y SUI 4 Son muy similares.

En la figura 5.15, la curva para 64QAM 2/3 cubre parcialmente a la curva para 64QAM 3/4, esto se debe a la parte aleatoria del modelo, es decir, en una simulación distinta donde los parámetros aleatorios tomen valores más favorables la distancia entre ambas curvas será mayor.

En la figura 5.17 se muestran las curvas BLER para 64QAM 3/4 en cada uno de los canales, lo que nos permite observar una aproximación al valor SNR donde el BLER cruza el umbral del rango de 10^{-2} , para cada canal. Dadas las características del canal SUI 1, es en este canal donde primero se llega al umbral deseado con SNR = 22 dB, mientras que la curva para el canal SUI 5, alcanza este punto alrededor de SNR = 43 dB, que es casi el doble del umbral del canal SUI 1, y que deja clara la diferencia entre el tipo de canal y el tipo de terreno para el que fueron diseñados.

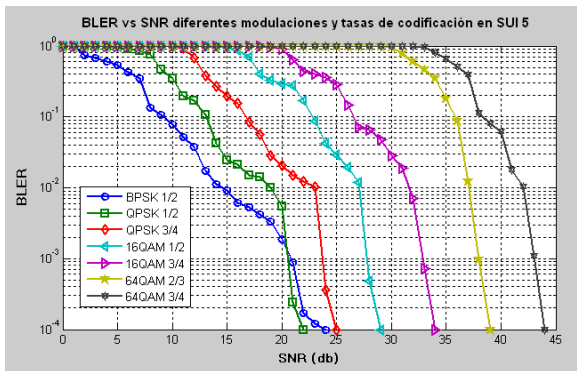


FIGURA 5.16: BLER vs SNR EN UN CANAL SUI 5.

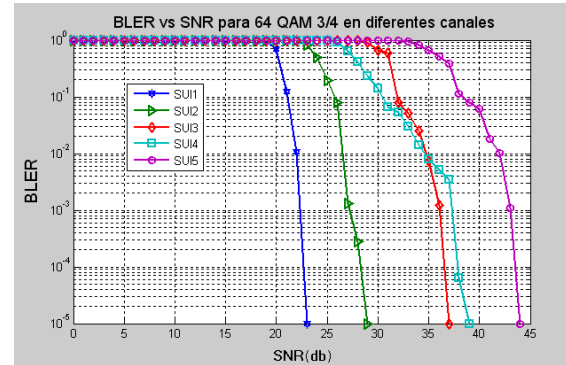


FIGURA 5.17: BLER vs SNR PARA 64QAM 3/4 PARA DIFERENTES CANALES SUI.

5.2.3 EFECTO DEL FORWARD ERROR CORRECTION

El *Forward Error Correction* es un método de control de errores en el que la fuente (emisor), envía los datos redundantes y el destino (receptor), sólo reconoce la parte de los datos que no contiene errores aparentes.

Como ya se ha mencionado antes, ver la sección 3.3.1.2, para esta tesis el FEC está formado por una concatenación del Reed- Solomon y código convolucional. La siguiente serie de gráficas, de la figura 5.18 a la figura 5.23, muestra el efecto que tiene el FEC sobre las curvas BER y BLER. Este efecto se representó gráficamente para 3 distintos esquemas de modulación en un canal SUI 2.

En estas gráficas es posible observar claramente los beneficios que implica un método de corrección de errores en general, y en particular el RS-CC sobre estas curvas.

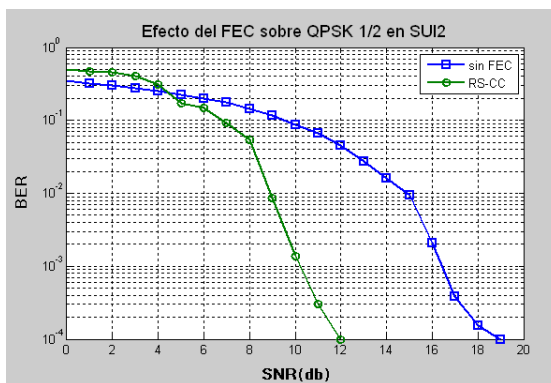


FIGURA 5.18: EFECTO DEL FEC SOBRE LA CURVA DE BER PARA QPSK 1/2 EN SUI2.

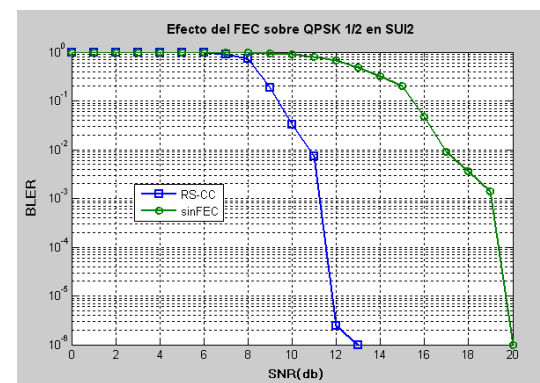


FIGURA 5.19: EFECTO DEL FEC SOBRE LA CURVA DE BER PARA QPSK 1/2 EN SUI2.

En estas gráficas se puede apreciar una diferencia de hasta 7 dB en las curvas con FEC y las curvas sin FEC para un canal SUI 1, sin embargo ello no implica que la diferencia entre implementar o no algún método de corrección de errores sea la misma para todos los canales y modulaciones, como ya se ha dicho esto depende, tanto en la simulación y, más aun en la realidad, de factores aleatorios.

La diferencia entre las curvas BER, con y sin FEC, para 16QAM $\frac{1}{2}$ en un canal SUI 2 es de 8 dB a favor del FEC, y para el mismo escenario las curvas del BLER muestran una diferencia de 6 dB. Para 64QAM $\frac{3}{4}$ la diferencia es aún mayor, tanto para el BER como para el BLER la diferencia está alrededor de los 10 dB. Todos estos valores pueden parecer mínimos, pero son la diferencia entre recibir la información correcta y percibir sólo ruido.

En todas estas gráficas es posible observar como las curvas sin FEC comienzan a descender primero, esto se debe a la cantidad de datos extra que implica el RS-CC, pero a medida que el SNR aumenta la utilidad de estos datos redundantes es obvia y favorable.

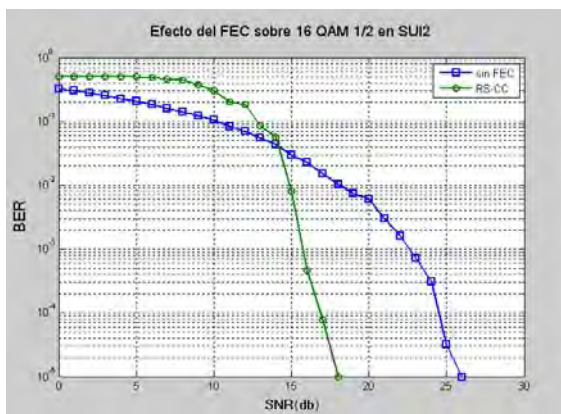


FIGURA 5.20: EFECTO DEL FEC SOBRE LA CURVA DE BER PARA 16QAM $\frac{1}{2}$ EN SUI2.

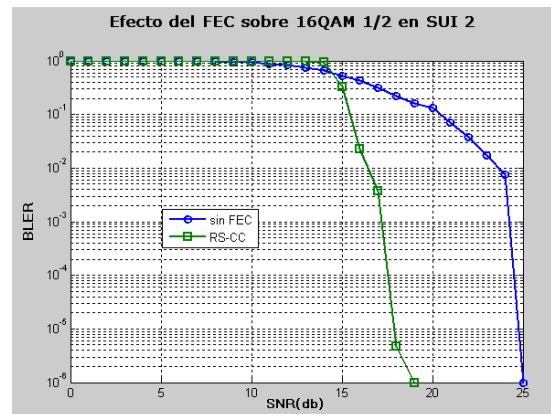


FIGURA 5.21: EFECTO DEL FEC SOBRE LA CURVA DE BLER PARA 16QAM $\frac{1}{2}$ EN SUI2.

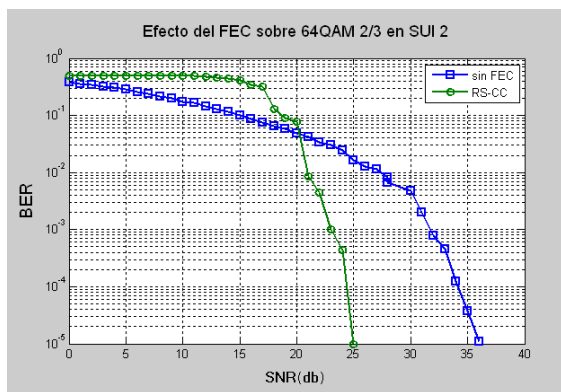


FIGURA 5.22: EFECTO DEL FEC SOBRE LA CURVA DE BER PARA 64QAM $\frac{1}{2}$ EN SUI2.

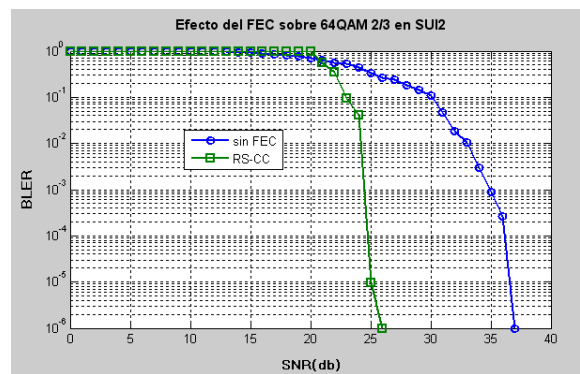


FIGURA 5.23: EFECTO DEL FEC SOBRE LA CURVA DE BLER PARA 64QAM $\frac{1}{2}$ EN SUI2.

5.2.4 EFECTO DE *INTERLEAVER*

El *Interleaver* distribuye los datos de una secuencia a otra distinta. El *interleaver* se utiliza en conjunto con la corrección de errores FEC.

El *Interleaver*, como se explica en la sección 3.3.1.3 de este proyecto, “revuelve” los datos, esto con el fin de que si una parte de los datos se daña durante la transmisión, al estar “revueltos” los bits de información dañados sean los menos o puedan recuperarse mediante los bits redundantes, generados por el FEC, de una sección no dañada.

Para ejemplificar el efecto del *interleaver* sobre la seguridad de la información, en las figuras 5.24 a 5.29, se presentan las curvas de BER y BLER, para tres escenarios distintos en un canal SUI 3: QPSK $\frac{3}{4}$, 16QAM $\frac{3}{4}$ y 64QAM $\frac{3}{4}$.

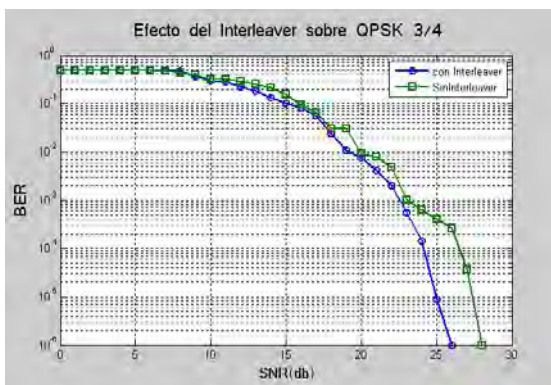


FIGURA 5.24: EFECTO DEL *INTERLEAVER* SOBRE LA CURVA DE BER PARA QPSK 3/4 EN SUI 3.

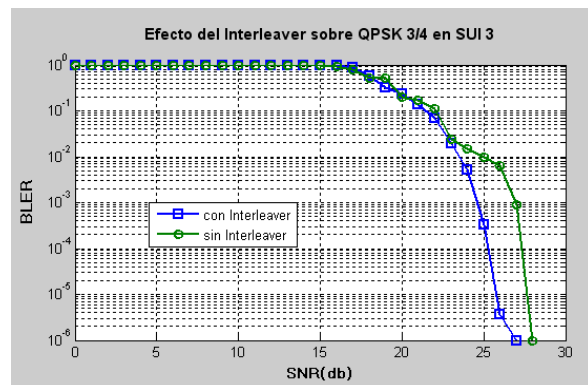


FIGURA 5.25: EFECTO DEL *INTERLEAVER* SOBRE LA CURVA DE BLER PARA QPSK 3/4 EN SUI 3.

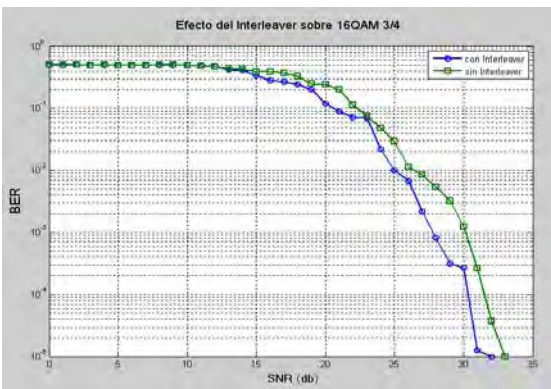


FIGURA 5.26: EFECTO DEL *INTERLEAVER* SOBRE LA CURVA DE BER PARA 16QAM 3/4 EN SUI 3.

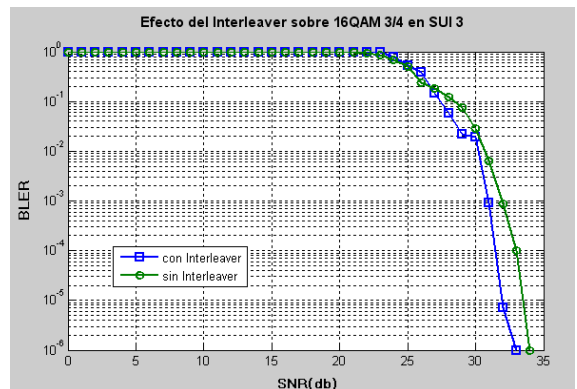


FIGURA 5.27: EFECTO DEL *INTERLEAVER* SOBRE LA CURVA DE BLER PARA 16QAM 3/4 EN SUI 3.

En contraste con la enorme diferencia entre las curvas con y sin FEC, la diferencia entre las curvas con o sin *interleaver* es más sutil, en las gráficas puede apreciar una diferencia de 1 o 2 dB para los tres escenarios, y aun cuando todos los elementos aleatorios tomaran los valores óptimos la diferencia no llegaría a ser tan clara como en las gráficas de la sección anterior.

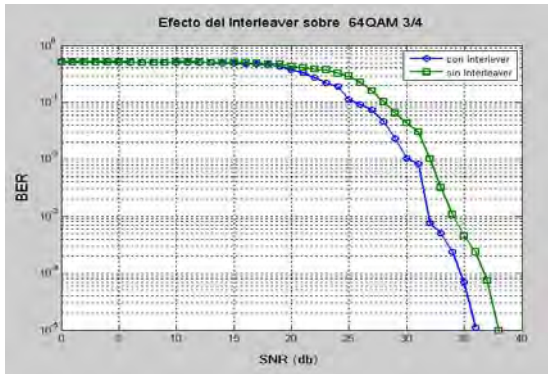


FIGURA 5.28: EFECTO DEL *INTERLEAVER* SOBRE LA CURVA DE BER PARA 64QAM 3/4 EN SUI 3.

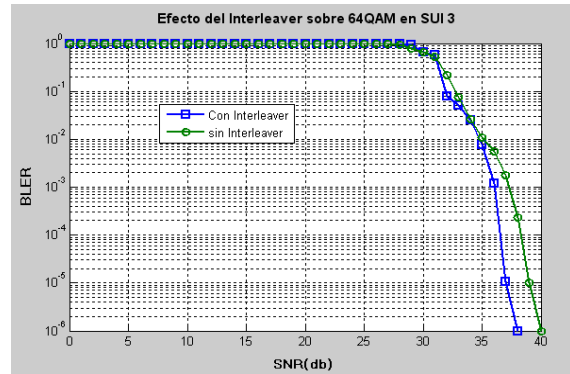


FIGURA 5.29: EFECTO DEL *INTERLEAVER* SOBRE LA CURVA DE BLER PARA 64QAM 3/4 EN SUI 3.

5.2.5 EFICIENCIA ESPECTRAL

La eficiencia espectral se refiere a la cantidad de información que puede transmitirse durante un determinado ancho de banda en un determinado sistema de comunicación. La existen varias formulas para calcular la eficiencia espectral, aunque en el afán de reproducir los resultados obtenidos en los textos de referencia definiremos la eficiencia espectral como:

$$\eta = (1-BER)^k m r \tag{5.2}$$

donde:

k es el tamaño de bloque de datos sin codificar;

m es el número de bits por símbolo y

r es la tasa de codificación.

En las figuras 5.30 a 5.33 se presentan las gráficas de eficiencia espectral, realizadas con los resultados obtenidos mediante las simulaciones, para los canales SUI 1 a 4.

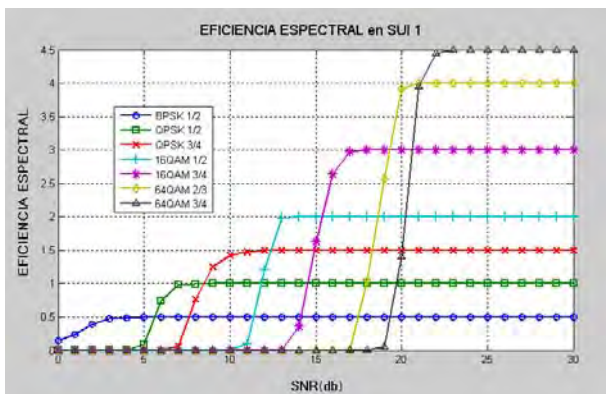


FIGURA 5.30: EFICIENCIA ESPECTRAL EN EL CANAL SUI 1.

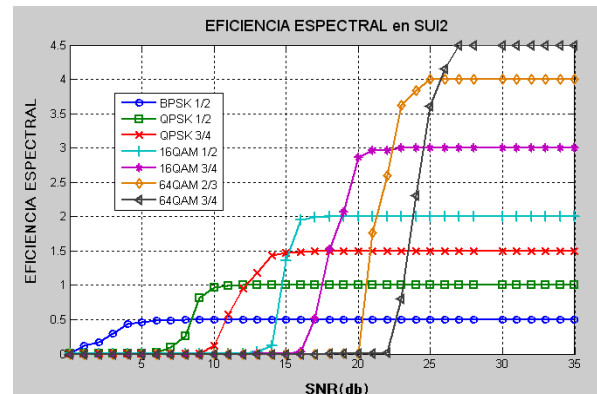


FIGURA 5.31: EFICIENCIA ESPECTRAL EN EL CANAL SUI 2.

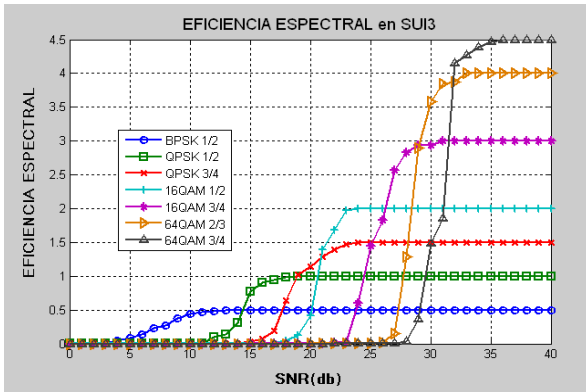


FIGURA 5.32: EFICIENCIA ESPECTRAL EN EL CANAL SUI 3.

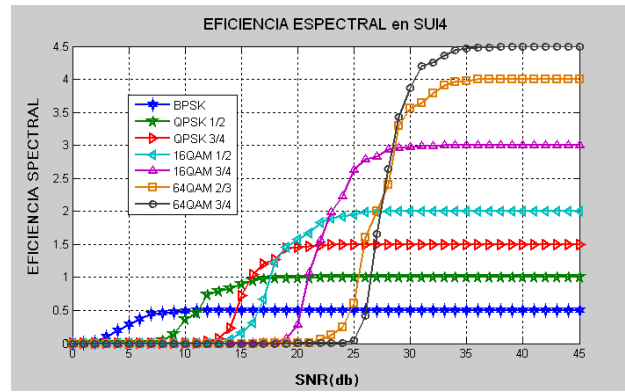


FIGURA 5.33: EFICIENCIA ESPECTRAL EN EL CANAL SUI 4.

En estas gráficas se observan nuevamente las similitudes y diferencias entre las curvas para el canal SUI3 y el SUI 4. Las curvas de eficiencia en el SUI 4 inician primero el ascenso pero es en el SUI 3 donde llegan primero al punto óptimo para cada escenario.

En la figura 5.34 se muestra la eficiencia espectral para el canal SUI 5 y, si se comparan estas curvas con las curvas de la figura 5.30, se puede observar que en algunos de los escenarios la eficiencia es mejor que en un SUI 4, esto se debe, además de a los elementos aleatorios favorables, al hecho que en este canal se utilizó un tiempo de guarda mayor, $G = 1/8$.

En la figura 5.35 presenta la eficiencia espectral para el esquema QPSK $3/4$ que permite observar más claramente la diferencia espectral entre los canales SUI 1, SUI 2 y SUI 4 para esta modulación,

Donde puede observarse como siempre la ventaja de la curva en SUI 1 respecto al resto de las curvas, pero dadas las características propias de dicho canal, ésto no es extraño.

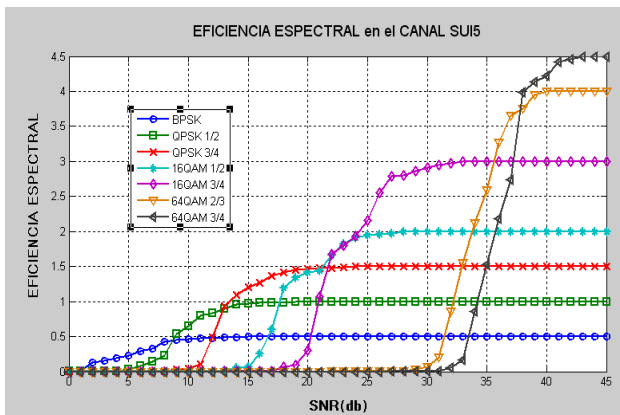


FIGURA 5.34: EFICIENCIA ESPECTRAL EN EL CANAL SUI 5.

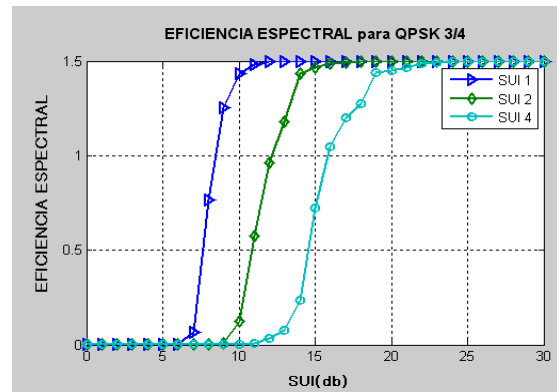


FIGURA 5.35: EFICIENCIA ESPECTRAL PARA QPSK $3/4$ EN VARIOS CANALES.

CONCLUSIONES

Las redes inalámbricas de banda ancha basadas en el estándar IEEE 802.16, bajo el nombre comercial de WiMAX, están siendo desarrolladas e implementadas alrededor del mundo. Un mundo globalizado donde la información lo es todo, y las premisas principales son tres: seguridad, velocidad y movilidad. El estándar IEEE 802.16-2004 ofrece dos de ellas, seguridad y velocidad, y en el estándar IEEE 802.16e agrega la tercera, movilidad.

La metodología empleada para investigar el rendimiento la capa física es la simulación. En este proyecto se muestran dos modelos que permiten simular la capa física de las redes fundamentadas en los estándares IEEE 802.16. Ambos modelos son herramientas útiles para diseñar y probar nuevos sistemas, con la ventaja de un bajo costo durante dicha implementación.

El modelo sencillo es más simple, y requiere de varias modificaciones estructurales para presentar resultados más apegados a la realidad. Una de estas modificaciones es la implementación de un "canal real", es decir algo más allá de un canal AWGN sencillo, ya que las variantes que afectan la información son más que sólo ruido Gaussiano, aun en el mejor de los escenarios.

El modelo compuesto es algo más complejo, sin embargo su estructura modular permite una mayor manipulación de los parámetros. En este modelo están implementados los seis modelos canales SUI, cinco de los cuales se simularon para este proyecto, lo que ofrece una visión más realista de los efectos del medio en la información. Las gráficas obtenidas mediante éstas y otras herramientas describen dichos efectos.

Las gráficas de dispersión que se generaron validan, en términos generales, los modelos en relación a la calidad media de la recepción, que varía según los distintos parámetros.

Entre las medidas clave del rendimiento de un sistema de comunicación están el BER y el BLER. El análisis de estas curvas permite definir un umbral a un rango de valores aceptables, que se traduce en velocidad.

La simulación de las curvas FEC, permiten el análisis de los efectos de la corrección de errores sobre la información y, en el caso del modelo compuesto, con relativa facilidad puede implementarse un FEC distinto al mostrado en este proyecto, RS-CC, y comparar cual ofrece una mejor tasa de corrección de errores.

Todo lo anterior se resume de la siguiente forma: Una buena herramienta de simulación ofrecerá una visión realista del desempeño de cualquier algoritmo, o sistema que se desee implementar en una red basada en el estándar IEEE 802.16. Y una buena herramienta permitirá pasar de un esquema fijo a uno móvil, con relativa facilidad. En el caso de los modelos descritos en este proyecto, el modelo compuesto es el más cercano a esta descripción.

Los valores arrojados por las simulaciones de ambos modelos, servirán como referencia para el desarrollo de nuevas herramientas de simulación, no sólo de capa física sino también de capas superiores, como la capa MAC.

Los resultados obtenidos en este proyecto serán una base para nuevos proyectos, algunos de los cuales se mencionan en la sección de Futuros Proyectos.

FUTUROS PROYECTOS

Los modelos de capa física descritos en este proyecto, pueden ser mejorados. Ambos modelos, el modelo compuesto y el modelo a bloques de *SIMULINK*, presentan varias opciones para futuros proyectos.

El modelo sencillo tiene una posibilidad muy obvia, la implementación de un canal distinto al canal AWGN actual. La implementación de uno o varios perfiles de canal sería un proyecto a futuro sobre este modelo. Y la implementación de un FEC distinto al simulado en esta versión del modelo ofrece otra opción.

El modelo compuesto, aun cuando es algo más elaborado que el modelo a bloques, requiere todavía mucho trabajo para poder ser la herramienta optima que se requiere. La parte de estimación de canal requiere más trabajo para lograr simular, con mayor exactitud, los efectos del canal sobre la información.

El estándar IEEE 802.16, en todas sus versiones, presenta muchas características optativas que pueden implementarse para mejorar el desempeño de una red fundamentada en este estándar, tanto en los modelos de simulación como en un sistema real.

Una de estas opciones es la aplicación de turbo códigos, BTC, para la parte de corrección de errores, o la implementación de STBC sobre el canal DL, con el fin de obtener transmisiones diversas.

ANEXO A

CÓDIGOS DE PROGRAMACIÓN EN *MATLAB* PARA EL MODELO *SIMULINK*

1. MODULACION QPSK

• Para CR $\frac{3}{4}$

```

clear;
input_data = sscanf(['45 29 C4 79 AD 0F 55 28 AD 87 B5 76 1A 9C 80 50 45 1B 9F D9 2A 88 95 EB AE B5 2E 03 4F 09 14 69 58 0A
5D'], '%x');
%-----
genPoly = gf(1,8);
for idx = 0 : 15
genPoly = conv(genPoly, [1 gf(2,8)^idx]);
end
primPoly = [1 0 0 0 1 1 1 0 1];
%-----
% Interleaving
% Ncpc= 2:QPSK, Ncbps: Número de bits a transmitir= 384
Ncbps = 384;
Ncpc = 2;
s = ceil(Ncpc/2);
% k: Número de bits -1 a transmitir, es el vector
k = 0 : Ncbps - 1;
% mk: Número de bits transmitidos después del primer ciclo y antes del segundo
mk = (Ncbps/12) * mod(k,12) + floor(k/12);
% jk: Número de bits transmitidos después del segundo ciclo
jk = s * floor(mk/s) + mod(s, mk + Ncbps - floor(12 * mk/Ncbps));
[s,int_idx]=sort(jk);
%-----
% QPSK Modulator
% Ncbps= 48x8bits=384; Ncpc=2; rate= 3/4; RS code (40,36,2); CC code 5/6;
% Uncoded block size= 36x8=288
Ry = ones(2,1)*[+1 -1];
ly = [+1 -1]*ones(1,2);
qamconst = Ry + j*ly;
qamconst = qamconst(:)/sqrt(2);

```

• Para CR $\frac{1}{2}$

```

clear;
input_data = sscanf(['45 29 C4 79 AD 0F 55 28 AD 87 B5 76 1A 9C 80 50 45 1B 9F D9 2A 88 95 EB AE B5 2E 03 4F 09 14 69 58 0A
5D'], '%x');
%-----
genPoly = gf(1,8);
for idx = 0 : 15
genPoly = conv(genPoly, [1 gf(2,8)^idx]);
end
primPoly = [1 0 0 0 1 1 1 0 1];
%-----

```

```

% Interleaving
% Ncpc= 2:QPSK, Ncbps: Número de bits a transmitir= 384
Ncbps = 384;
Ncpc = 2;
s = ceil(Ncpc/2);
% k: Número de bits -1 a transmitir, es el vector
k = 0 : Ncbps - 1;
% mk: Número de bits transmitidos después del primer ciclo y antes del segundo
mk = (Ncbps/12) * mod(k,12) + floor(k/12);

% jk: Número de bits transmitidos después del segundo ciclo

jk = s * floor(mk/s) + mod(s, mk + Ncbps - floor(12 * mk/Ncbps));
[s,int_idx]=sort(jk);
%-----
% %QPSK Modulator
% Ncbps= 48x8bits=384; Ncpc=2; rate= 1/2; RS code (32,24,4); CC code 2/3;
% Uncoded block size= 24x8=192
Ry = ones(2,1)*[+1 -1];
ly = [+1 -1]*ones(1,2);
qamconst = Ry + j*ly;

qamconst = qamconst(:)/sqrt(2);

```

2. 16 - QAM PARA CR $\frac{3}{4}$

```

input_data = sscanf(['45 29 C4 79 AD 0F 55 28 AD 87 B5 76 1A 9C 80 50 45 1B 9F D9 2A 88 95 EB AE B5 2E 03 4F 09 14 69 58 0A 5D'], '%x');
%-----
genPoly = gf(1,8);
for idx = 0 : 15
genPoly = conv(genPoly, [1 gf(2,8)^idx]);
end
primPoly = [1 0 0 0 1 1 1 0 1];
%-----
% Interleaving
% Ncpc= 4:16QAM, Ncbps: Número de bits a transmitir = 768
Ncbps = 768;
Ncpc = 4;
s = ceil(Ncpc/2);s = ceil(Ncpc/2);
% k: Número de bits -1 a transmitir, es el vector
k = 0 : Ncbps - 1;
% mk: Número de bits transmitidos después del primer ciclo y antes del segundo
mk = (Ncbps/12) * mod(k,12) + floor(k/12);
% jk: Número de bits transmitidos después del segundo ciclo
jk = s * floor(mk/s) + mod(s, mk + Ncbps - floor(12 * mk/Ncbps));
[s,int_idx]=sort(jk);
%-----
% 16 QAM
% Ncbps= 96x8bits=768; Ncpc=4; rate= 3/4; RS code (80,72,4); CC code 5/6
% Uncoded block size= 72x8=576
Ry = ones(4,1)*[+1 +3 -1 -3];
ly = [+1 +3 -1 -3]*ones(1,4);
qamconst = Ry + j*ly;
qamconst = qamconst(:)/sqrt(10);

```

3. 64 - QAM PARA CR $\frac{3}{4}$

```

(genPoly, [1 gf(2,8)^idx]);
end
primPoly = [1 0 0 0 1 1 1 0 1];
%-----
% Interleaving
% Ncpc= 6:64QAM, Ncbps: Número de bits a transmitir = 1152
Ncbps = 1152;
Ncpc = 6;
s = ceil(Ncpc/2);
% k: Número de bits -1 a transmitir, es clear;
input_data = sscanf(['45 29 C4 79 AD 0F 55 28 AD 87 B5 76 1A 9C 80 50 45 1B 9F D9 2A 88 95 EB AE B5 2E 03 4F 09 14 69 58 0A 5D'], '%x');
%-----
genPoly = gf(1,8);
for idx = 0 : 15
genPoly = convex vector
k = 0 : Ncbps - 1;
% mk: Número de bits transmitidos después del primer ciclo y antes del segundo
mk = (Ncbps/12) * mod(k,12) + floor(k/12);
% jk: Número de bits transmitidos después del segundo ciclo
jk = s * floor(mk/s) + mod(s, mk + Ncbps - floor(12 * mk/Ncbps));
[s,int_idx]=sort(jk);
%-----
% 64 QAM
% Ncbps= 144x8bits=1152; Ncpc=6; rate= 3/4; RS code (120,108,6); CC code 5/6
% Uncoded block size= 108x8=864
Ry = ones(8,1)*[+3 +1 +5 +7 -3 -1 -5 -7];
ly = [+3 +1 +5 +7 -3 -1 -5 -7]*ones(1,8);
qamconst = Ry + j*ly;
qamconst = qamconst(:)/sqrt(42);

```

ANEXO B

CÓDIGO *MATLAB* PARA EL TRASMISOR Y RECEPTOR *IEEE 802.16* Y LOS MODELOS SUI DE CANAL

- **Parámetros *IEEE 802.16*.** Parámetros globales de la simulación.

```
function[IEEE 80216params]=IEEE 80216_params(G, BWe, SUI, TD)

IEEE 80216params.Ofdm.BW=BWe;
IEEE 80216params.channel.P=1;
IEEE 80216params.Ofdm.Nused=200;
BW=IEEE 80216params.Ofdm.BW/(10^6);
if (rem(BW,1.75)==0)
    IEEE 80216params.Ofdm.n=8/7;
else if (rem(BW,1.5)==0)
    IEEE 80216params.Ofdm.n=86/75;
    else if (rem(BW,1.25)==0)
        IEEE 80216params.Ofdm.n=144/125;
    else if (rem(BW,2.75)==0)
        IEEE 80216params.Ofdm.n=316/275;
    else if(rem(BW,2.0)==0)
        IEEE 80216params.Ofdm.n=57/50;
    else %otherwise
        IEEE 80216params.Ofdm.n=8/7;
    end
end
end
end
end
IEEE 80216params.Ofdm.G=G;%1/4; % ratio of CP time to useful time
%%% derived OFDM symbol parameter
IEEE 80216params.Ofdm.Nfft=256; % smallest power of 2 greater than Nused
%sampling frequency
IEEE 80216params.Ofdm.Fs=floor((IEEE 80216params.Ofdm.n*IEEE
80216params.Ofdm.BW)/8000)*8000;
%subcarrier spacing
IEEE 80216params.Ofdm.del_f=IEEE 80216params.Ofdm.Fs/IEEE 80216params.Ofdm.Nfft;
%useful symbol time
IEEE 80216params.Ofdm.Tb=1/(IEEE 80216params.Ofdm.del_f);
%CP time
IEEE 80216params.Ofdm.Tg=IEEE 80216params.Ofdm.G*IEEE 80216params.Ofdm.Tb;
%symbol time
IEEE 80216params.Ofdm.SymbolTime=IEEE 80216params.Ofdm.Tg+IEEE 80216params.Ofdm.Tb;
%sampling time
IEEE 80216params.Ofdm.SampleTime=IEEE 80216params.Ofdm.Tb/IEEE 80216params.Ofdm.Nfft;
%number of pilot carrier
IEEE 80216params.Ofdm.Npilot=8;
%number of data carrier
IEEE 80216params.Ofdm.Ndata=IEEE 80216params.Ofdm.Nused- IEEE 80216params.Ofdm.Npilot;
%MAPPING Parameters : full subchannelization
IEEE 80216params.Map.DataSubCars=[29:40 42:65 67:90 92:115 117:128 130:141 143:166
168:191 193:216 218:229];
IEEE 80216params.Map.PilotSubCars=[41 66 91 116 142 167 192 217];
IEEE 80216params.Map.GuardSubCars=[1:28 230:256];
IEEE 80216params.Map.UsedSubCars=[29:128 130:229];
```



```

% Link Parameters
%*****
if (TD==1)
    IEEE 80216params.Link.direction='Dlink';
else
    if (TD==2)
        IEEE 80216params.Link.direction='Ulink';
    end
end
%*****
%always model applies to this
IEEE 80216params.Link.DIUC=7;
IEEE 80216params.Link.BSID=1;
IEEE 80216params.Link.FrameNo=1; % transmission start from frame number 1
IEEE 80216params.Link.Frames=250; % number of frames to be sent
IEEE 80216params.Link.FrameTime=4*10^(-3);
IEEE 80216params.Link.Alloc_frac=0.1;
IEEE 80216params.Link.BurstTime=IEEE 80216params.Link.Alloc_frac*IEEE
80216params.Link.FrameTime;
%
% RS encoder parameter
IEEE 80216params.RS.generator=rsgenpoly(255,239,[],0); %RS field and code generator
%CC encoder parameter
IEEE 80216params.CC.trellis=poly2trellis(7,[171 133]);% CC trellis as per specification
IEEE 80216params.CC.tblen=32; %the traceback length
%MODULATION params
IEEE 80216params.Modulation.subchn=16; %full subchannelization
%gray coded mappings
graymap_16=[ '1101'
             '1100'
             '1110'
             '1111'
             '1001'
             '1000'
             '1010'
             '1011'
             '0001'
             '0000'
             '0010'
             '0011'
             '0101'
             '0100'
             '0110'
             '0111'];
graymap_64=[ '111011'
             '111010'
             '111000'
             '111001'
             '111101'
             '111100'
             '111110'
             '111111'
             '110011'
             '110010'
             '110000'
             '110001'
             '110101'
             '110100'
             '110110'
             '110111'
             '100011'
             '100010'
             '100000'
             '100001'

```

```

'100101'
'100100'
'100110'
'100111'
'101011'
'101010'
'101000'
'101001'
'101101'
'101100'
'101110'
'101111'
'001011'
'001010'
'001000'
'001001'
'001101'
'001100'
'001110'
'001111'
'000011'
'000010'
'000000'
'000001'
'000101'
'000100'
'000110'
'000111'
'010011'
'010010'
'010000'
'010001'
'010101'
'010100'
'010110'
'010111'
'011011'
'011010'
'011000'
'011001'
'011101'
'011100'
'011110'
'011111'];
% binary to decimal conversion
mapping_16qam=bin2dec(graymap_16).';
mapping_64qam=bin2dec(graymap_64).';
mapping_qpsk= [ 2 3 0 1];
clear graymap_16;
clear graymap_64;
%QPSK - implemented as 4 QAM
constell_qpsk=qammod((0:3),4);%corchetes por parentesis
IEEE 80216params.Modulation.gray_map_qpsk(mapping_qpsk(:)+1)=constell_qpsk(:);
%16QAM
constell_16qam=qammod((0:15),16);%corchetes por parentesis
IEEE 80216params.Modulation.gray_map_16qam(mapping_16qam(:)+1)=constell_16qam(:);
%64QAM
constell_64qam=qammod((0:63),64);%corchetes por parentesis
IEEE 80216params.Modulation.gray_map_64qam(mapping_64qam(:)+1)=constell_64qam(:);
clear mapping_qpsk;
clear mapping_16qam;
clear mapping_64qam;
clear constell_qpsk;
clear constell_16qam;

```

```

clear constell_64qam;
IEEE 80216params.simOpts.TxDiv=1;
%*****
%disp(IEEE 80216params.Ofdm.G);
%*****
IEEE 80216params.simOpts.RxDiv=1;
IEEE 80216params.simOpts.DoppTaps=256; % taps in Doppler filter
%*****
if (SUI==1)
    IEEE 80216params.simOpts.ChanModel='SUI1';
else if (SUI==2)
    IEEE 80216params.simOpts.ChanModel='SUI2';
else if (SUI==3)
    IEEE 80216params.simOpts.ChanModel='SUI3';
else if (SUI==4)
    IEEE 80216params.simOpts.ChanModel='SUI4';
else if (SUI==5)
    IEEE 80216params.simOpts.ChanModel='SUI5';
else (SUI==6)
    IEEE 80216params.simOpts.ChanModel='SUI6';
end
end
end
end
%*****
%IEEE 80216params.simOpts.ChanModel='SUI2';
switch(IEEE 80216params.simOpts.ChanModel)
case 'SUI1'
    IEEE 80216params.channel.P=[0 -15 -20];
    IEEE 80216params.channel.K=[4 0 0];
    IEEE 80216params.channel.tau=[0 0.4 0.9];
    IEEE 80216params.channel.Dop=[0.4 0.3 0.5];
    IEEE 80216params.channel.AntCorr=0.7;
    IEEE 80216params.channel.Fnorm=-0.1771;
    IEEE 80216params.channel.Trms=0.111;
case 'SUI2'
    IEEE 80216params.channel.P=[0 -12 -15];
    IEEE 80216params.channel.K=[2 0 0];
    IEEE 80216params.channel.tau=[0 0.4 1.1];
    IEEE 80216params.channel.Dop=[0.2 0.15 0.25];
    IEEE 80216params.channel.AntCorr=0.5;
    IEEE 80216params.channel.Fnorm=-0.393;
    IEEE 80216params.channel.Trms=0.202;
case 'SUI3'
    IEEE 80216params.channel.P=[0 -5 -10];
    IEEE 80216params.channel.K=[1 0 0];
    IEEE 80216params.channel.tau=[0 0.4 0.9];
    IEEE 80216params.channel.Dop=[0.4 0.3 0.5];
    IEEE 80216params.channel.AntCorr=0.4;
    IEEE 80216params.channel.Fnorm=-1.5113;
    IEEE 80216params.channel.Trms=0.264;
case 'SUI4'
    IEEE 80216params.channel.P=[0 -4 -8];
    IEEE 80216params.channel.K=[0 0 0];
    IEEE 80216params.channel.tau=[0 1.5 4];
    IEEE 80216params.channel.Dop=[0.2 0.15 0.25];
    IEEE 80216params.channel.AntCorr=0.3;
    IEEE 80216params.channel.Fnorm=-1.9218;
    IEEE 80216params.channel.Trms=1.257;
case 'SUI5'
    IEEE 80216params.channel.P=[0 -5 -10];
    IEEE 80216params.channel.K=[0 0 0];
    IEEE 80216params.channel.tau=[0 4 10];

```

```

IEEE 80216params.channel.Dop=[2 1.5 2.5];
IEEE 80216params.channel.AntCorr=0.3;
IEEE 80216params.channel.Fnorm=-1.5113;
IEEE 80216params.channel.Trms=2.842;
case 'SUI6'
IEEE 80216params.channel.P=[0 -10 -14];
IEEE 80216params.channel.K=[0 0 0];
IEEE 80216params.channel.tau=[0 14 20];
IEEE 80216params.channel.Dop=[0.4 0.3 0.5];
IEEE 80216params.channel.AntCorr=0.3;
IEEE 80216params.channel.Fnorm=-0.5683;
IEEE 80216params.channel.Trms=5.240;
end
%
%%Fading
IEEE 80216params.channel.P=10.^(IEEE 80216params.channel.P/10);%dB to linear scale
IEEE 80216params.channel.variance=IEEE 80216params.channel.P./(IEEE
80216params.channel.K+1);%variance
IEEE 80216params.channel.meanp=IEEE 80216params.channel.P.*(IEEE
80216params.channel.K./(IEEE 80216params.channel.K+1));
IEEE 80216params.channel.m=sqrt(IEEE 80216params.channel.meanp);
IEEE 80216params.channel.paths_c=IEEE 80216params.channel.m;
IEEE 80216params.channel.Nyquist_freq=2*max(IEEE 80216params.channel.Dop);% Nyquist freq.
of channel
IEEE 80216params.channel.Nyquist_time=(1/IEEE 80216params.channel.Nyquist_freq);
IEEE 80216params.channel.conv_locs=1+round(IEEE 80216params.channel.tau/(IEEE
80216params.Ofdm.SampleTime*10^6));
%%channel filter
IEEE 80216params.channel.filter=zeros(length(IEEE 80216params.channel.P),IEEE
80216params.simOpts.DoppTaps);
for i=1:length(IEEE 80216params.channel.P)
D=IEEE 80216params.channel.Dop(i)/max(IEEE 80216params.channel.Dop)/2;
f0=(0:floor(IEEE 80216params.simOpts.DoppTaps*D))/(floor(IEEE
80216params.simOpts.DoppTaps*D));%corchetes por parentesis
PSD = 0.785*f0.^4 - 1.72*f0.^2 + 1.0;
filter=[PSD(1:end-1) zeros(1,IEEE 80216params.simOpts.DoppTaps-2*floor(IEEE
80216params.simOpts.DoppTaps*D)) PSD(end:-1:2) ];
filter=sqrt(filter);
filter=ifftshift(ifft(filter));
filter= real(filter);
filter=filter/sqrt(sum(filter.^2));
IEEE 80216params.channel.filter(i,:)=filter;
end

IEEE 80216params.channel.Normfact=10^(IEEE 80216params.channel.Fnorm/20);
%save IEEE 80216params;

```

- **Randomizer.** Aleatorización de datos.

```

function [randomized_data] = randomizer(datain, IEEE 80216params)

data = datain;
if (IEEE 80216params.Link.DIUC==0) && (IEEE 80216params.Link.direction=='Dlink')
seed_value=[0 0 0 0 0 0 0 1 0 1 0 1 0 0 1];
else
seed1=de2bi(IEEE 80216params.Link.BSID,4,'left-msb');
seed2=de2bi(IEEE 80216params.Link.DIUC,4,'left-msb');
seed3=de2bi(IEEE 80216params.Link.FrameNo,4,'left-msb'); %The frame number
seed_value= horzcat(seed1,horzcat([1 1],horzcat(seed2,horzcat(1,seed3))));
end;
for i=1:size(data,2)
xor_out= bitxor(seed_value(15), seed_value(14));
randomized_data(i)= bitxor(xor_out, data(i));

```

```

        seed_value=[xor_out seed_value(1:14)];
end
randomized_data;
disp('RANDOMIZED DATA');
clear seed_value
clear data

```

- **Tipo de datos.** Define el tipo de datos y la tasa de codificación.

```

function [ch_cod_prof_no]=tipo_dato(n_mod_tipo, tasa)

switch n_mod_tipo
    case 1
        disp('BPSK 1/2');
        ch_cod_prof_no = 1;
    case 2
        if tasa==(2/3)
            disp('QPSK 2/3');
            ch_cod_prof_no = 2;
        elseif tasa == (5/6)
            disp('QPSK 5/6');
            ch_cod_prof_no = 3;
        end
    case 3
        if tasa == (2/3)
            disp('16QAM 2/3');
            ch_cod_prof_no = 4;
        elseif tasa == (5/6)
            disp('16QAM 5/6');
            ch_cod_prof_no = 5;
        end
    case 4
        if tasa == (3/4)
            disp('64QAM 3/4');
            ch_cod_prof_no = 6;
        elseif tasa == (5/6)
            disp('64QAM 5/6');
            ch_cod_prof_no = 7;
        end
end
end

```

- **channel coding profile.** Define parámetros en función del tipo de dato y la tasa de codificación.

```

%%% channel coding profile selection
function [IEEE_80216params]=ch_coding_profile(ch_cod_prof_no,IEEE_80216params)
switch(ch_cod_prof_no)
case 1
    IEEE_80216params.Modulation.Type='BPSK';
    IEEE_80216params.RS.N=12;
    IEEE_80216params.RS.K=12;
    IEEE_80216params.RS.T=0;
    IEEE_80216params.CC.p=1;
    IEEE_80216params.CC.q=2;
case 2
    IEEE_80216params.Modulation.Type='QPSK';
    IEEE_80216params.RS.N=32;
    IEEE_80216params.RS.K=24;
    IEEE_80216params.RS.T=4;

```

```

        IEEE 80216params.CC.p=2;
        IEEE 80216params.CC.q=3;
case 3
        IEEE 80216params.Modulation.Type='QPSK';
        IEEE 80216params.RS.N=40;
        IEEE 80216params.RS.K=36;
        IEEE 80216params.RS.T=2;
        IEEE 80216params.CC.p=5;
        IEEE 80216params.CC.q=6;
case 4
        IEEE 80216params.Modulation.Type='16QAM';
        IEEE 80216params.RS.N=64;
        IEEE 80216params.RS.K=48;
        IEEE 80216params.RS.T=8;
        IEEE 80216params.CC.p=2;
        IEEE 80216params.CC.q=3;
case 5
        IEEE 80216params.Modulation.Type='16QAM';
        IEEE 80216params.RS.N=80;
        IEEE 80216params.RS.K=72;
        IEEE 80216params.RS.T=4;
        IEEE 80216params.CC.p=5;
        IEEE 80216params.CC.q=6;
case 6
        IEEE 80216params.Modulation.Type='64QAM';
        IEEE 80216params.RS.N=108;
        IEEE 80216params.RS.K=96;
        IEEE 80216params.RS.T=6;
        IEEE 80216params.CC.p=3;
        IEEE 80216params.CC.q=4;
case 7
        IEEE 80216params.Modulation.Type='64QAM';
        IEEE 80216params.RS.N=120;
        IEEE 80216params.RS.K=108;
        IEEE 80216params.RS.T=6;
        IEEE 80216params.CC.p=5;
        IEEE 80216params.CC.q=6;
otherwise
        error('Not a valid profile number: valid range[1:6]');
end
%based on this profile decide on the data length
Nosyms=floor(IEEE 80216params.Link.BurstTime/IEEE 80216params.Ofdm.SymbolTime);
IEEE 80216params.Link.DataLeng=IEEE 80216params.RS.K*Nosyms -1; %number of bytes
IEEE 80216params.Link.DataLeng=IEEE 80216params.RS.K*Nosyms -1; %number of bytes
%save IEEE 80216params;

```

- **RS encoder.** Codificación REED-SOLOMON

```

function [rs_encoded_data]= rs_encoder(randomized_data, IEEE 80216params)
data = randomized_data;
%*****RS_encoder*****
generator=IEEE 80216params.RS.generator;
N=IEEE 80216params.RS.N;
K=IEEE 80216params.RS.K;
T=IEEE 80216params.RS.T;
%Data manipulation for RS CODER Input
num_bits=size(data,2); % number of bit in data block
num_bytes=num_bits/8;% number of byte
bytes =(bi2de(reshape(data,8,num_bytes).','left-msb').');
num_blocks=ceil(num_bytes/K);

if(num_blocks==floor(num_bytes/K))

```

```

        num_blocks=num_blocks+1;
end

bytes(num_bytes+1:num_blocks*K-1)=255;

%last byte in the bust is 0x00
bytes=[bytes 0];
%now do the encoding
disp('aquí');
msg_block=reshape(bytes,K,num_blocks).'; %the rows are the blocks to be encoded
%RS encoding is bypassed for BPSK modulation
if(N == K)
    rs_data=msg_block;
else
    %do RS encoding for other scheme
    %rsenc_block=rsenc(gf(msg_block,8),N,K,[],'beginning');
    %rs_data=double(rsenc_block.x); % conversion of GF into Double

    %UNAM: Se agreea polinomio generalizador y polinomio generalizador
    %Además el msg_block se convierte a una cadenas de 239 bytes para que
    %pueda utilizarse con Reed - Solomon.

    msg_block_239 = [zeros(num_blocks,255 -K -16) msg_block];
    rsenc_255=rsenc(gf(msg_block_239,8),255,239,generator,'beginning');
    rsenc_block = [rsenc_255(1:N-K) rsenc_255(255-K+1:255)];
    rs_data=double(rsenc_block.x); % conversion of GF into Double
end
%conversion to binary
num_blocks=size(rs_data,1);
%disp('rs_data');
%disp(rs_data);
for i=1:num_blocks
    %get the binary data from decimal numbers
    bit_data=de2bi(rs_data(i,:),'8','left-msb').';
    rs_encoded_data(i,:)=bit_data(:)';
end
%rs_encoded_data = [];
%disp(rs_encoded_data);
%xlswrite('data',rs_encoded_data.','Hoja 1','B2');
%xlswrite('Ulink1',rs_encoded_data.','Hoja 1','B2');
clear rs_data;
clear bit_data;
clear data;
%disp(' ');

```

- **CC Encoder.** Codificación convolucional

```

function conv_encoded_data=conv_encoder(rs_encoded_data, IEEE_80216params)

data = rs_encoded_data;
p=IEEE_80216params.CC.p;
q=IEEE_80216params.CC.q;
%determine number of blocks
num_blocks = size(data,1)
%CC encoding of each block
for i=1:num_blocks
    conv_encoded_data(i,:)=convenc(data(i,:),IEEE_80216params.CC.trellis);
end
clear num_blocks;
if (p==2)&&(q==3) %X1Y1Y2:
    conv_encoded_data(:,3:4:end)=[];

```

```

else
    if (p==3)&&(q==4) %X1Y1Y2Y3
        conv_encoded_data(:,3:6:end)=[];
        conv_encoded_data(:,5:5:end)=[];
    else
        if (p==5)&&(q==6) %X1Y1Y2X3Y4X5
            conv_encoded_data(:,3:10:end)=[];
            conv_encoded_data(:,5:9:end)=[];
            conv_encoded_data(:,5:8:end)=[];
            conv_encoded_data(:,7:7:end)=[];
        end
    end
end
%xlswrite('Ulink1',conv_encoded_data.','Hoja 1','c2');
clear p;
clear q;
clear data;

```

- **Interleaver**

```

function [interleaved_data] = interleaver(conv_encoded_data,IEEE_80216params)

data = conv_encoded_data;
switch (IEEE_80216params.Modulation.Type)
    case 'BPSK'
        Ncbps= 12* IEEE_80216params.Modulation.subchn;
        s= ceil( 1/2 );
    case 'QPSK'
        Ncbps= 24* IEEE_80216params.Modulation.subchn;
        s= ceil( 2/2 );
    case '16QAM'
        Ncbps= 48* IEEE_80216params.Modulation.subchn;
        s= ceil( 4/2 );
    case '64QAM'
        Ncbps= 72* IEEE_80216params.Modulation.subchn;
        s= ceil( 6/2 );
end
%%check
if ((size(data,2) < Ncbps) || (size(data,2) > Ncbps))
    error('size mismatch');
end
% first permutation accoring to eqn. 71
for k=0:Ncbps-1
    mk=(Ncbps/12)*mod(k,12)+floor(k/12);
    firstPerm_interleaved_data(:,mk+1)=data(:,k+1);
end
clear k;
clear mk;
clear punct_code;
%second permutation according to eqn. 72
for k=0:Ncbps-1
    jk=s*floor(k/s)+mod((k+Ncbps-floor(12*k/Ncbps)),s);
    interleaved_data(:,jk+1)=firstPerm_interleaved_data(:,k+1);
end
%xlswrite('Ulink1',interleaved_data.','Hoja 1','d2')
clear jk;
clear data;
clear firstPerm_interleaved_data;
clear Ncbps;

```


- pilot modulator

```
function [w_k]=pilot_modulator(interleaved_data, IEEE 80216params)

data = interleaved_data;
sequence_length=size(data,1);
if (IEEE 80216params.Link.direction == 'Dlink')
    initialization_seq=de2bi(hex2dec('7FF'),'left-msb');
    Symbol_off=2;
else
    initialization_seq= de2bi(hex2dec('555'),'left-msb');
    Symbol_off=1;
end
for i=1:(sequence_length+Symbol_off)
    initialization_seq_msb=bitxor(initialization_seq(11),initialization_seq(9));
    initialization_seq=[initialization_seq_msb initialization_seq(1:10)];
    w_k(i)=initialization_seq_msb
end
w_k(1:Symbol_off)=[];
```

- constalletaion mapper

```
function [const_mapped_data]= constellation_mapper(interleaved_data, w_k, IEEE
80216params)

data = interleaved_data;
num_blocks=size(data,1);% get the block size, in bits, that gets encoded
const_mapped_data=zeros(num_blocks,IEEE 80216params.Ofdm.Nfft);%initialize
const_mapped_data to zero
clear num_blocks;
switch (IEEE 80216params.Modulation.Type)
case 'BPSK'
    modulated_data=pskmod(data,2);
case 'QPSK'
    symbol_size=2;
    scaling_fact= sqrt(1/2);
    %convert the symbol into [0...M-1]
    for i=1:symbol_size:size(data,2)
        mod_inp(:,floor(i/symbol_size) +1)=bi2de(data(:,i:i+symbol_size-1),'left-
msb');
    end
    %QPSK is implemented as 4 QAM
    %scaled modulated data
    modulated_data=scaling_fact*genqammod(mod_inp,IEEE
80216params.Modulation.gray_map_qpsk);
case '16QAM'
    symbol_size=4;
    scaling_fact= sqrt(1/10);
    %convert the symbol into [0...M-1]
    for i=1:symbol_size:size(data,2)
        mod_inp(:,floor(i/symbol_size) +1)=bi2de(data(:,i:i+symbol_size-1),'left-
msb');
    end
    %scaled modulated data
    modulated_data=scaling_fact*genqammod(mod_inp,IEEE
80216params.Modulation.gray_map_16qam);
case '64QAM'
    symbol_size=6;
    scaling_fact= sqrt(1/42);
    %convert the symbol into [0...M-1]
    for i=1:symbol_size:size(data,2)
        mod_inp(:,floor(i/symbol_size) +1)=bi2de(data(:,i:i+symbol_size-1),'left-
msb');
```

```

        end
        %scaled modulated data
        modulated_data=scaling_fact*genqammod(mod_inp,IEEE
80216params.Modulation.gray_map_64qam);
    end
        % place the modulated data into data subcarriers
        const_mapped_data(:,IEEE 80216params.Map.DataSubCars)=modulated_data(:,1:end);
clear data;
clear modulated_data;
%fill in the pilot subcarriers
if (IEEE 80216params.Link.direction=='Dlink')
    const_mapped_data(:,41)=complex(1-2*w_k,0);
    const_mapped_data(:,91)=complex(1-2*w_k,0);
    const_mapped_data(:,192)=complex(1-2*w_k,0);
    const_mapped_data(:,217)=complex(1-2*w_k,0);
    const_mapped_data(:,66)=complex(1-2*(1-w_k),0);
    const_mapped_data(:,116)=complex(1-2*(1-w_k),0);
    const_mapped_data(:,142)=complex(1-2*(1-w_k),0);
    const_mapped_data(:,167)=complex(1-2*(1-w_k),0);
else %Ulink
    const_mapped_data(:,41)=complex(1-2*w_k,0);
    const_mapped_data(:,91)=complex(1-2*w_k,0);
    const_mapped_data(:,192)=complex(1-2*w_k,0);
    const_mapped_data(:,217)=complex(1-2*w_k,0);
    const_mapped_data(:,142)=complex(1-2*w_k,0);
    const_mapped_data(:,167)=complex(1-2*w_k,0);
    const_mapped_data(:,66)=complex(1-2*(1-w_k),0);
    const_mapped_data(:,116)=complex(1-2*(1-w_k),0);
end
end

```

- ofdm modulator

```

function [timedomain_data_vec] = ofdm_modulator(const_mapped_data, IEEE 80216params)

data = const_mapped_data;
timedomain_data=ifft(data. ');
%CP length
CP_len=IEEE 80216params.Ofdm.G*size(timedomain_data,1);
% append the CP at the beginning of time data
timedomain_data_cp=[timedomain_data(end+1-CP_len:end,:);timedomain_data];
% time domain data vector
timedomain_data_vec=timedomain_data_cp(:). ';
%IEEE 802.16 Receiver

```

- ch_Fading

```

function paths_r=ch_Fading(leng,IEEE 80216params)

ch_no= IEEE 80216params.simOpts.TxDiv*IEEE 80216params.simOpts.RxDiv; % number of antenna
pairs
L=length(IEEE 80216params.channel.P); % number of taps
%% coeff. generation
for j=1:ch_no
paths_r(j, :, :)=sqrt(1/2)*randn(L, leng)+j*randn(L, leng).*((sqrt(IEEE
80216params.channel.variance))'*ones(1, leng));
    for i=1:L
        temp(1:leng)=paths_r(j, i, :);
        path=fftfilt(IEEE 80216params.channel.filter(i, :), [temp zeros(1, IEEE
80216params.simOpts.DoppTaps)]);%corchetes por parentesis
    end
end

```

```

        paths_r(j,i,:)=path(1+IEEE 80216params.simOpts.DoppTaps/2:end-IEEE
80216params.simOpts.DoppTaps/2);
    end
end
%%%% Correlation Matrix
for i=1:ch_no
    for j=1:ch_no
        if (i~=j)
            correlation_matrix(i,j)=IEEE 80216params.channel.AntCorlnFac;
        else
            correlation_matrix(i,j)=1;
        end
    end
end
correlation_matrix=sqrtm(correlation_matrix);
% correlate according to correlation matrix
for tap=1:L
    paths_r(:,tap,:)=correlation_matrix*squeeze(paths_r(:,tap,:));
end
for i=1:ch_no
    paths_r(i,:,:)=squeeze(paths_r(i,:,:));
end
for tap=1:L
    paths_r(1:end,tap,1:end)=IEEE
80216params.channel.paths_c(tap)+paths_r(1:end,tap,1:end);
end
% coeff. normalization
paths_r=IEEE 80216params.channel.Normfact*paths_r;

```

- **cir**

```

function [CIR,time]=cir(coeffs,time,systeme, IEEE 80216params)

persistent counter;
if (isempty(counter) || (systeme==0))
    counter=1;
end
ch_no= IEEE 80216params.simOpts.TxDiv*IEEE 80216params.simOpts.RxDiv; % number of antenna
pairs
if(systeme-time >=IEEE 80216params.channel.Nyquist_time)
    counter=mod(counter,length(coeffs))+1;
    time=time+IEEE 80216params.channel.Nyquist_time;
end
% initialize the CIR with zeros
CIR=zeros(ch_no,max(IEEE 80216params.channel.conv_locs));
%generate CIR
for i=1:length(IEEE 80216params.channel.tau)
    CIR(:,IEEE 80216params.channel.conv_locs(i))=CIR(:,IEEE
80216params.channel.conv_locs(i))+coeffs(:,i,counter);
end

```

- **SUI- Channel**

```

function [rx_data]=channel_sui(tx_data,ch_impulse_resp,IEEE 80216params)

rx_data=zeros(IEEE 80216params.simOpts.RxDiv,size(tx_data,2)+size(ch_impulse_resp,2)-1);

for i=1:IEEE 80216params.simOpts.RxDiv
    for j=1:IEEE 80216params.simOpts.TxDiv
        rx_data(i,:)=rx_data(i,:)+conv(tx_data(j,:),ch_impulse_resp((i-1)*IEEE
80216params.simOpts.TxDiv+j,:));
    end
end

```

```

end
end

leng=size(rx_data,2);
disp('leng');
disp(leng);
%%AWGN NOISE
noise_varience=IEEE 80216params.Ofdm.Nfft/IEEE 80216params.Ofdm.Nused/(10^(IEEE
80216params.simOpts.SNR/10))/2;
noise=sqrt(noise_varience)*(randn(IEEE 80216params.simOpts.RxDiv,leng)+j*randn(IEEE
80216params.simOpts.RxDiv,leng));
%% adding noise to rx_data
rx_data=rx_data+noise;

```

- ofdm demodulator

```

function [data_sub,pilot_sub]=ofdm_demodulator(rx_signal, IEEE 80216params)
symbol_length=IEEE 80216params.Ofdm.Nfft*(1+IEEE 80216params.Ofdm.G);%symbol length
no_of_symbols=floor(size(rx_signal,2)/symbol_length);%number of symbol
for i=1:IEEE 80216params.simOpts.RxDiv
    rx_data=rx_signal(i,:);
    clear rx_signal;
    ofdm_symbol=rx_data(1:no_of_symbols*symbol_length);
    ofdm_symbol=reshape(ofdm_symbol,symbol_length,no_of_symbols);
    clear rx_data;
    % separating guard
    ofdm_symbol=ofdm_symbol(symbol_length-IEEE 80216params.Ofdm.Nfft+1:symbol_length,:);
    %fft operation
    freq_domain_data=fft(ofdm_symbol)/(IEEE 80216params.Ofdm.Nfft/sqrt(IEEE
80216params.Ofdm.Nused)/IEEE 80216params.simOpts.RxDiv);
    %separation of pilot and data symbol
    data_sub(i,:,:)=freq_domain_data(IEEE 80216params.Map.DataSubCars,:);
    pilot_sub(i,:,:)=freq_domain_data(IEEE 80216params.Map.PilotSubCars,:);
end
%save data_sub;
%save pilot_sub;

```

- de-mapper

```

function [demod_bit_stream]=demodulator(ofdm_demod_symbol,IEEE 80216params)

switch (IEEE 80216params.Modulation.Type)
case 'BPSK'
    %There is no need for scaling in BPSK
    demodulated_symbol=pskdemod(ofdm_demod_symbol,2);
    symbol_size=1;
case 'QPSK'
    %scaling
    scalin_fact=sqrt(1/2);
    ofdm_demod_symbol=ofdm_demod_symbol/scalin_fact;
    %4QAM demodulation
    demodulated_symbol=genqamdemod(ofdm_demod_symbol,IEEE
80216params.Modulation.gray_map_qpsk);
    symbol_size=2;
case '16QAM'
    scalin_fact=sqrt(1/10);
    ofdm_demod_symbol=ofdm_demod_symbol/scalin_fact;
    %16QAM demodulation
    demodulated_symbol=genqamdemod(ofdm_demod_symbol,IEEE
80216params.Modulation.gray_map_16qam);
    symbol_size=4;

```

```

case '64QAM'
    scalin_fact=sqrt(1/42);
    ofdm_demod_symbol=ofdm_demod_symbol/scalin_fact;
    %64QAM demodulation
    demodulated_symbol=genqamdemod(ofdm_demod_symbol,IEEE
80216params.Modulation.gray_map_64qam);
    symbol_size=6;
end
disp(ofdm_demod_symbol);
%symbol to bit conversion
s=size(demodulated_symbol,2);
for i=1:s
    demodulated_bit=de2bi(demodulated_symbol(:,i),symbol_size,'left-msb');
    demod_bit_stream(:,i)=demodulated_bit(:);

    %disp('ggg');
    %disp(demodulated_bit);
    %disp(demod_bit_stream);
end
demod_bit_stream=demod_bit_stream(:).';

```

- **de-interleaver**

```

function[deinterleaved_data] = deinterleaver(demod_bit_stream,IEEE 80216params)

data = demod_bit_stream;

switch (IEEE 80216params.Modulation.Type)
    case 'BPSK'
        Ncbps= 12* IEEE 80216params.Modulation.subchn;
        s= ceil( 1/2 );
    case 'QPSK'
        Ncbps= 24*IEEE 80216params.Modulation.subchn;
        s= ceil( 2/2 );
    case '16QAM'
        Ncbps= 48* IEEE 80216params.Modulation.subchn;
        s= ceil( 4/2 );
    case '64QAM'
        Ncbps= 72* IEEE 80216params.Modulation.subchn;
        s= ceil( 6/2 );
end
% primer permutacion
for j=0:Ncbps-1
    disp('s');
    disp(s);
    jk=s*floor(j/s)+mod(j+floor(12*j/Ncbps),s)+1;
    disp('jk');
    disp(jk);
    firstperm_deinterleaved_data(:,jk)= data(:,j+1);
end
% segunda permutacion
for j=0:Ncbps-1
    j1=12*j-(Ncbps-1)*floor(12*j/Ncbps)+1;
    deinterleaved_data(:,j1)=firstperm_deinterleaved_data(:,j+1);
end

```

- **De Coder**

```

function convdecod_data=conv_decoder(deinterleaved_data,IEEE 80216params)

data = deinterleaved_data;

```

```

p=IEEE 80216params.CC.p;
q=IEEE 80216params.CC.q;

data=-2*data +1;
if ((p==1) && (q==2)) %Puncturing is not required
    punc_pattern=[1 2];
    s=1*2;
else
if ((p==2) && (q==3))
%X1Y1Y2
punc_pattern=[1 2 4];
s=2*2;
else
if ((p==3) && (q==4))
%X1Y1Y2X3
punc_pattern=[1 2 4 5];
s=2*3;
else
%X1Y1Y2X3Y4X5
if ((p==5) &&(q==6))
punc_pattern=[1 2 4 5 8 9];
s=2*5;
end
end
end
end
syms=size(data,1);
rem_size=rem(size(data,2), length(punc_pattern));
for i=1:syms
    depunct=zeros(s,floor(size(data,2)/length(punc_pattern)));
    depunct_bits=reshape(data(i,1:end-
rem_size),length(punc_pattern),floor(size(data,2)/length(punc_pattern)));
    depunct(punc_pattern,:)=depunct(punc_pattern,:)+depunct_bits;
    if(rem_size ~=0)
    remd=zeros(1,s);
    remd_bits=data(end-rem_size+1:end);
    remd(punc_pattern(1:rem_size))=remd_bits+remd(punc_pattern(1:rem_size));
    native_code(i,:)=[depunct(:)' remd];
    else
    native_code(i,:)=depunct(:)';
    end
end
end
%viterbi decoding of native_code
for i=1:syms
    convdecod_data(i,:)=vitdec(native_code(i,:),IEEE
80216params.CC.trellis,96,'trunc','unquant');
end
%xlswrite('Ulink1',convdecod_data.','Hoja 1','B3');

```

- **rs_decoder**

```

function [rsdecoded_data errs_corr]=rs_decoder(convdecod_data,IEEE 80216params)

data = convdecod_data;
generator=IEEE 80216params.RS.generator;
N=IEEE 80216params.RS.N;
K=IEEE 80216params.RS.K;
T=IEEE 80216params.RS.T;

%RS decoder starts here
syms=size(data,1);% number bits
num_bytes=size(data,2)/8; % number of bytes in each block
for i=1:syms

```

```
%bit to byte conversion
    bytes = bi2de(reshape(data(i,:),8,num_bytes).','leftmsb').');
    dBlock(i,:)=bytes;
end
if(N == K)% bypass RS decoding for BPSK
    rsdecoded_data=dBlock;
    errs_corr=0;
else
%decode the data and get the error correction count
    [rsdecoded_data,errs_corr]=rsdec(gf(dBlock,8),N,K,generator,'beginning');
    % GF to double conversion
    rsdecoded_data=double(rsdecoded_data.x);
end
```

GLOSARIO

AAS	<i>Adaptive Antenna System.</i>
ADC	<i>Analog to Digital Conversion.</i> Conversión analógica digital.
ADSL	<i>Asymmetric Digital Subscriber Line.</i> Línea de suscripción asimétrica digital.
AES	<i>Advanced Encryption Standard.</i>
ARIB	<i>Association of Radio Industries and Businesses.</i>
ATM	<i>Asynchronous Transfer Mode.</i> Modo de transferencia asíncrono.
AWGN	<i>Additive White Gaussian Noise.</i>
BCH	<i>Bose-Chaudhuri-Hocquenghem.</i>
BER	<i>Bit Error Rate.</i> Tasa (razón) de bits errados.
BLER	<i>Block Error Rate.</i> Tasa (razón) de bloques errados.
Bluetooth SIG	<i>Bluetooth Special Interest Group.</i>
BPSK	<i>Binary Phase Shift Keyin.</i> Modulación por desplazamiento en fase binaria.
BS	<i>Base Station.</i> Estación base.
BSID	<i>Base Station ID.</i>
BTC	<i>Block Turbo Coding.</i> Bloque de turbo código.
BWA	<i>BroadBand Wireless Access.</i> Acceso inalámbrico de banda ancha.
CC	<i>Convolutional Code.</i> Código Convolutacional.
CCK	<i>Complimentary Code Keying.</i>
CDMA-EVDO	<i>Code Division Multiple Access - Evolution-Data Optimized .</i>
CID	<i>Identificadores de conexión.</i>
CP	<i>Cyclic Prefix.</i> Prefijo cíclico.
CPE	<i>Customer Premises Equipment.</i> Equipos locales del cliente.
CPS	<i>Common Part Sublayer.</i> Parte común de la subcapa.
CS	<i>Convergence Sublayer.</i> Subcapa de convergencia.
CSMA	<i>Carrier Sense Multiple Access.</i>
CSMA/CA	<i>CSMA/ Collision Avoidance.</i>

CSMA/CD	<i>CSMA/ Collision Detection.</i>
CTC	Convolutional Turbo Codec.
DAC	Digital to Analog Conversion. Conversión digital analógica.
DAMA	<i>Demand Assignment Multiple Acces.</i> Acceso múltiple con asignación en función a la demanda.
DCD	<i>Descripción del canal DL.</i>
DCF	<i>Distributed Coordination Function.</i>
DFS	Dynamic Frequency Selection. Selección dinámica de frecuencia.
DFT	Discrete Fourier Transform. Transformada discreta de Fourier.
DIUC	<i>Downlink Interval Usage Code.</i>
DL	<i>Downlink.</i>
DL-MAP	Downlink Map.
DSL	<i>Digital Subscriber Line.</i> Línea suscriptor Digital.
DSSS	Direct Sequence-Spread Spectrum.
EDCA	<i>Enhanced Distributed Channel Access.</i>
EDGE	<i>Enhanced Data rates for GSM Evolution.</i>
ETSI	<i>European Telecommunications Standards Institute.</i> Instituto europeo de estándares de telecomunicaciones.
FCH	<i>Frame Control Header.</i>
FDD	Frequency Division Duplexing. Dúplex por división de frecuencia.
FDM	Frequency Division Multiplexing. Multiplexación por división de frecuencia.
FEC	<i>Forward Error Correction.</i> Corrección de errores hacia adelante.
FFT	<i>Fast Fourier Transform.</i> Transformada rápida de Fourier.
Gbps	Giga bits por segundo.
GFSK	Gaussian Frequency Shift Keying.
GPRS	<i>General Paket Radio Service.</i>
GPS	<i>Global Positioning System.</i>
GSM	<i>Generic Routing Encapsulation.</i>
HCCA	<i>HCF Controlled Channel Access.</i>

HCF	<i>Hybrid Coordination Function.</i>
HyperLAN2	<i>High Performance Radio Local Area Network Type 2.</i>
HiperMAN	<i>High PERFORMANCE Metropolitan Area Network.</i> Red de área metropolitana de alto desempeño.
HISWANA	<i>High-Speed Wireless Local Area Network.</i>
IAPP	<i>Inter-Access Point Protocol.</i>
ICI	<i>InterCarrier Interference.</i> Interferencia inter- portadora.
IDFT	<i>Inverse Discrete Fourier Transform.</i> Inversa de la Transformada de discreta Fourier.
IEEE	<i>Institute of Electrical and Electronics Engineers.</i> Instituto de Ingenieros Eléctricos y Electrónicos.
IP	<i>Internet Protocol.</i>
ISI	<i>InterSymbol Interference.</i> Interferencia inter-símbolo.
ISM	Industrial, Scientifical, Medical.
Kbps	<i>Kilo bits por segundo.</i>
LAN	<i>Local Area Network.</i> Red de área local.
LOS	<i>Line of Sight.</i> Línea de vista.
MAC	<i>Medium Access Control.</i> Control de acceso al medio.
MAN	<i>Metropolitan Area Network.</i>
Mbps	Mega bits por segundo.
MCM	<i>MultiCarrier Modulation.</i> Modulación multi- portadora.
MIMO	Multiple Input Multiple Output.
MSB	Bits más significativos.
N_{cbps}	<i>Número de bits codificados por subcanal.</i>
NLOS	<i>Non Line of Sight.</i> No línea de vista. Sin línea de vista.
OFDM	<i>Orthogonal Frequency Division Multiplexing.</i> Multiplexación por división de frecuencia ortogonal.
OFDMA	<i>Orthogonal Frequency Division Multiple Access.</i> Acceso múltiple por división de frecuencia ortogonal.
PBCC	<i>Packet Binary Convolutional Coding.</i>
PCI	<i>Peripheral Component Interconnect.</i> Conexión de componentes periféricos.

PDA	Personal Digital Assistant.
PDU	Paket Data unit.
PHY	PHYSical layer.
PMP	<i>Point to Multipoint</i> . Punto a multipunto.
POS	<i>Point of Sale</i> . Espacio personal.
PRBS	<i>Pseudo Random Bit Sequencer</i> . Secuencia binaria pseudo-aleatoria.
PSK	<i>PreShared Key</i> .
PSD	<i>Power Spectral Density</i> .
PTP	<i>Point to Point</i> . Punto a punto.
QAM	<i>Quadrature Amplitude Modulation</i> . Modulaci3n de amplitud en cuadratura.
QoS	<i>Quality of Service</i> . Calidad de servicio.
QPSK	<i>Quadrature PhaseShift keying</i> . Modulaci3n por desplazamiento de fase en cuadratura.
RF	<i>Radio Frequency</i> .
RMS	<i>Root-Mean-Square</i> . Retardo de propagaci3n.
RS	<i>Reed-Solomon</i> .
SIG	<i>Special Interest Group</i>
SDMA	<i>Spatial Division Multiple Access</i> .
SDU	<i>Service Data Unit</i> .
SNR	<i>Signal to Noise Ratio</i> . Relaci3n se1al a ruido.
SOFDMA	<i>Scalable Orthogonal Frequency Division Multiplexing Access</i> .
SS	<i>Subscriber Stations</i> . Estaci3n subscriptora.
SSID	Service Set Identifier. Identificador de una red WLAN.
STBC	<i>Space Time Block Code</i> .
STC	<i>Space Time Coding</i> .
SUI	<i>Stanford University Interim</i> .
TDD	<i>Time Division Duplexing</i> . D3plex por divisi3n de tiempo.
TDM	<i>Time Division Multiplexing</i> . Multiplexaci3n por divisi3n de tiempo.
TDMA	<i>Time Division Multiple Access</i> . Acceso m3ltiple por divisi3n de tiempo.

TKIP	<i>Temporal Key Integrity Protocol.</i>
TTG	<i>Transmit / Receive Transition Gap.</i>
UCD	Descripción del canal UL.
UIUC	<i>Uplink Interval Usage Code.</i>
UL	<i>Uplink.</i>
UL-MAP	<i>Uplink Map.</i>
UTMS	<i>Universal Mobile Telecommunication System.</i>
VoIP	<i>Voice over IP. Voz sobre IP.</i>
WAN	<i>Wide Area Network. Redes de área extensa.</i>
WCDMA	<i>Wideband Code Division Multiple Access.</i>
WEP	<i>Wired Equivalent Privacy.</i>
WI-FI	<i>Wireless Fidelity.</i>
WiMAX	<i>Worldwide Interoperability for Microwave Access.</i>
WirelessMAN	<i>Wireless Metropolitan Area Network. Red inalámbrica de área metropolitana.</i>
WLAN	<i>Wireless LAN.</i>
WPA	Wi-Fi Protected Access.
WPAN	WirelessPHysical layer.

REFERENCIAS

- [1] IEEE 802.16-2004, "IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed BroadBand Wireless Access Systems". October 2004.
- [2] IEEE Std 802.16a, "Local and metropolitan area networks Part 16: Air Interface for Fixed BroadBand Wireless Access Systems". April 08, 2003.
- [3] Nadeem Khan, Muhammad "WiMAX Physical Layer Implementation in Simulink". En Linea [URL]: <http://www.mathworks.com/Matlabcentral/fileexchange/loadFile.do?objectId=15513&ref=rssfeed&id=mostDownloadedFiles>. 2007.
- [4] M. A. Hasan, "Performance Evaluation of WiMAX/IEEE 802.16 OFDM Physical Layer", Tesis de Maestria, Helsinki University of Technology, Espoo, EN, Finlandia, 2007.
- [5] C. Barnes, D. Zendzian, J. Posluns, E. Ouellet, D. Lloyd y T. Bautts, "Hack Proofing Your Wireless Network", Rockland, MA: **Syngress Publishing, Inc.** 2002.
- [6] M. Gast, "802.11 Wireless Networks: The Definitive Guide", **O'REYLY**. 2002.
- [7] J. Khan y A. Khwaja, "Building Secure Wireless Networks with 802.11", West Sussex: **John Wiley & Sons Ltd.** 2003.
- [8] P. Nikipolitis, M.S. Obaidat, G.I. Papadimitriou y A.S. Pomportsis, "WIRELESS NETWORKS", West Sussex: **John Wiley & Sons Ltd.** 2003.
- [9] J. Andrews, A. Ghosh, y R. Muhamed, "Fundamentals of WiMAX: understanding broadband wireless networking", Massachusetts: **Prentice Hall.** 2007
- [10] Kwang-Cheng Chen y J. R. B. De Marca, "Mobile WiMAX" West Sussex: **John Wiley & Sons Ltd.** 2008.
- [11] F. Ohrtman, y K. Roeder, "Wi-Fi Handbook: Building 802.11b Wireless Networks", Nueva York: **McGraw-Hill.** 2003.
- [12] A. Tanenbaum, "Computer Networks", New Jersey: **Pearson Education Inc.** 2002.
- [13] F. Ohrtman, "WiMAX Handbook Building 802.16 Wireless Networks", Nueva York: **McGraw-Hill.** 2005.
- [14] R. Montañana, "Acceso Residencial de Banda Ancha Inalámbricos", Departamento de Informática, Universidad de Valencia, invierno 2007.
- [15] IEEE 802.16.3c-01/29r4: *Channel Models for Fixed Wireless Applications* 2001-07-16
- [16] C.P. Hoymann, B. Walke y B. M. Wolz, "WiMAX Metropolitan Area Networks and its Future Potentials", Chair of Communication Networks (ComNets), RWTH Aachen University, Aachen, Alemania, 2006. En Linea [URL]: <http://www.ikr.uni-stuttgart.de/eunice2006/WiMAXTutorial.pdf#search=%22WiMAX%20ieee%20802.16%20mesh%20filetype%3Apdf%22>
- [17] R. Flickenger, "Wireless Networking in the Developing World", ebook: <http://wndw.net>. 2007.
- [18] R. Olexa, "Implementing 802.11, 802.16, and 802.20 Wireless Networks", Burlington, MA: **Elsevier Inc.** 2005.
- [19] S.S Miller, "Wi-Fi Security", Nueva York: **McGraw-Hill.** 2003.

WEB PAGES

- [w1] Unión Internacional de Telecomunicaciones, "*Tendencias globales en las telecomunicaciones*", En Línea [URL]: <http://www.itu.int/itu-news/manager/display.asp?lang=es&year=2007&issue=07&ipage=Telecom-trends&ext=html>
- [w2] QuiteRoot Solutions, "*Clasificarea retelelor de calculatoare*", 2007. En línea [URL]: <http://www.quietroot.org/resurse/documente/doc/Retele-de-calculatoare.html>.
- [w3] Roland Piquepaille's Technology Trends, "*Las Vegas High-Speed Mesh Network*", En Línea [URL]: <http://www.primidi.com/2005/01/04.html>. 2005
- [w4] Álvarez Férez, A; Cañero Morera, V. "Codificación Convolutiva y Decodificación Viterbi – Algoritmos", En Línea [URL]: <http://ma.alvarez0005.eresmas.net/trabajos/ccvsatelite/teoria.html>
- [w5] Metrologic México, "*Estándares Inalámbricos*", 2008. En línea [URL]: http://www.metrologicmexico.com/contenido1/informacion_tecnica/estandares_inalambricos.php.
- [w6] Romo Zamudio, José Fabián. "*WiMAX. Lo inalámbrico a larga distancia*", En línea [URL]: <http://www.enterate.unam.mx/Articulos/2005/abril/WiMAX.htm>
- [w7] Andrews, J; Ghosh, A; y Muhamed, R. "*The nuts and bolts of WiMAX*", 2007. En Línea [URL]: <http://www.mobilehandsetdesignline.com/201500260;jsessionid=PQGI3TZSVSMIGQSNDLQCKH0CJUNN2JVN?printableArticle=true>.
- [w8] Info-citel, "*Redes Inalámbricas de Banda Ancha*", 2006. En línea [URL]: http://www.citel.oas.org/newsletter/2006/marzo/banda-ancha_e.asp.
- [w9] "The MatWorks" En Línea [URL]: <http://www.mathworks.com/>