



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA – TELECOMUNICACIONES

“IMPLEMENTACIÓN DE UN TRANSMISOR 802.11 OFDM BASADO EN SDR”

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
GUSTAVO RAFAEL LUNA PORTILLA

TUTOR
DR. VÍCTOR RANGEL LICEA. FACULTAD DE INGENIERÍA

MÉXICO, D. F. AGOSTO 2015

JURADO ASIGNADO:

Presidente: Dr. José María Matías Maruri

Secretario: Dr. Miguel Moctezuma Flores

Vocal: Dr. Víctor Rangel Licea

1^{er.} Suplente: Dr. Esau Vicente Vivas

2^{d o.} Suplente: Dr. Ramón Gutiérrez Castrejón

Lugar o lugares donde se realizó la tesis: México, D.F.

TUTOR DE TESIS:

Dr. Víctor Rangel Licea

FIRMA





Agradecimientos

A la UNAM

Por darme la oportunidad de estudiar, aprender y mejorar como ser humano, al permitirme ser parte ella y de su sociedad.

Al Dr. Víctor Rangel Licea

Por fungir como mi asesor y apoyarme a lo largo de mis estudios de posgrado, permitiéndome integrarme a su equipo de investigación.

Al CONACYT

Por el apoyo económico que me otorgó para poder realizar mi maestría.

A la DGAPA-UNAM

Por el apoyo otorgado para la realización de esta tesis por parte del proyecto de investigación PAPIIT No. IN114713.

Al Dr. Robert Morelos Zaragoza

Por su apoyo en la estancia realizada en la San José State University, y los conocimientos que nos transmitió para mejorar los resultados de esta tesis.

A mi padre

Por ser el apoyo incondicional y forjador ser la persona que soy, siendo el principal pilar que me permitió llegar a concluir mis estudios de maestría.





Resumen

El estándar de Wi-Fi, es decir, 802.11 fue creado en 1997, para definir una serie de parámetros de construcción que los creadores de tecnologías debían seguir, en caso de utilizar dispositivos Wi-Fi en sus productos. La estandarización principalmente se creó para que existiera una interoperabilidad entre dispositivos.

Las exigencias de la sociedad moderna, obligan a que el estándar tenga que actualizarse constantemente, obligando a los investigadores a buscar nuevas soluciones, que permitan realizar cambios a tecnologías de una manera menos costosa, sin la necesidad de cambiar el hardware, por ejemplo con el uso de Software Defined Radio.

En la presente tesis se describe la implementación de un transmisor IEEE 802.11g mediante Software Defined Radio, utilizando los equipos SR N210 y el lenguaje de programación LabVIEW. El código desarrollado se comparó con el ejemplo del estándar IEEE 802.11, para comprobar su correcto funcionamiento y además se realizaron distintas pruebas, para comprobar su fiabilidad como sistema comercial. También se decidió crear el código de sincronización de un transmisor IEEE 802.11g, ya que los elementos necesarios para realizar una detección de símbolo correcta, son dados en los frames de transmisión.

En base a los resultados obtenidos en las pruebas de Bit Error Rate en distintos escenarios, se comprobó la fiabilidad del uso de equipos USRP N210 y LabVIEW, para una implementación comercial. Adicionalmente se comprobó la máxima velocidad de transmisión que se puede lograr con las características de equipos utilizados, otorgando un modelado experimental para aplicaciones Wi-Fi con los equipos USRP-N210.





Índice General

Índice de figuras

Índice de tablas

1. Introducción	1
1.1. Antecedentes	1
1.2. Definición del problema	2
1.3. Objetivos y metas	3
1.4. Contribución y relevancia	4
1.5. Estructura de la tesis	4
2. Marco teórico	6
2.1. Redes wi-fi	6
2.1.1. Historia del estándar IEEE 802.11	6
2.1.2. Versiones del estándar IEEE 802.11	6
2.2. LabVIEW	9
2.2.1. Descripción general de LabVIEW	9
2.2.2. Programación en LabVIEW.	9
2.2.3. Uso del USRP N210 para transmisión en LabVIEW	12
2.3. Estado del arte	15
2.3.1. IEEE 802.11 wireless LAN implemented on software defined radio with hybrid programmable architecture	15
2.3.2. Performance evaluation for OFDM link based on LabVIEW and USRP.	16



2.3.3. IEEE 802.11p transmission using GNURadio	16
2.3.4. An IEEE 802.11a/g/p OFDM in Receiver for GNURadio	17
2.3.5. An IEEE 802.11 MAC software defined radio Implementation for experimental wireless communications and networking research.	18
3. Estándar IEEE 802.11g	20
3.1. Aspectos básicos del estándar IEEE 802.11	20
3.1.1. Topología IEEE 802.11.	20
3.1.2. Esquemas de transmisión	22
3.2. Subcapa MAC.	25
3.2.1. Carrier sense multiple access with collision avoidance	25
3.2.2. Distributed coordination function	26
3.2.3. Tipos de frame	31
3.3. Capa física	34
3.3.1. Descripción del frame de capa física	35
3.3.2. Procesamiento de los datos	40
4. USRP N210	45
4.1. Características generales.	45
4.2. Motherboard y daughterboard	48
4.3. Conexión con el USRP N210.	50
5. Descripción del código	53
5.1. Frame de capa MAC	53
5.1.1. Valores iniciales.	54



5.1.2. CRC-32.	55
5.2. Frame de capa PHY	57
5.2.1. Selector de modulación	57
5.2.2. Padding	59
5.2.3. Bit de paridad	59
5.3. Scrambler.	60
5.4. Codificación convolucional	61
5.5. Puncturing	62
5.6. Interleaver	63
5.7. Modulador	64
5.8. Pilotos.	66
5.9. IFFT y ciclo prefijo	68
5.10. Secuencias de entrenamiento.	70
5.11. Conexión con el USRP y herramientas para un posterior análisis .	71
5.12. Sincronización.	73
6. Pruebas y resultados	75
6.1. Descripción del equipo utilizado	76
6.2. Obtención de la frecuencia a transmitir.	80
6.3. Prueba de sincronía y constelaciones	81
6.4. Medición del espectro.	85
6.5. Prueba de bit error rate a 1m de distancia	86
6.6. Prueba de distancia contra BER	90



6.6.2. Distancia vs BER en interiores	90
6.6.2. Distancia vs BER en exteriores.	94
Conclusiones	97
Trabajo a futuro	99
Anexo 1	100
Lista de términos	106
Bibliografía	109





Índice de Figuras

Figura 2.1 Ejemplo de un panel frontal en LabVIEW	10
Figura 2.2 Ejemplo de un diagrama a bloques en LabVIEW	10
Figura 2.3 Bloque de un control de un arreglo entero en el panel frontal y en el diagrama a bloques	11
Figura 2.4 Bloque de un indicador de un arreglo complejo en el panel frontal y en el diagrama a bloques	11
Figura 2.5 niUSRP Open Tx Session VI	12
Figura 2.6 niUSRP Configure Signal VI	13
Figura 2.7 niUSRP Write Tx Data (poly) VI	14
Figura 2.8 niUSRP Close Session VI	14
Figura 3.1 Topología de la norma IEEE 802.11	21
Figura 3.2 Comparación de uso de ancho de banda de FDM contra OFDM	23
Figura 3.3 Esquema de un Transmisor OFDM	24
Figura 3.4 Esquema de un Receptor OFDM	24
Figura 3.5 Envío de un frame con CSMA/CA	26
Figura 3.6 Estación A oculta de C	27
Figura 3.7 Censado virtual utilizando RTS/CTS	28
Figura 3.8 Representación en el tiempo de los IFS del estándar IEEE 802.11g	30
Figura 3.9 Estructura del frame de datos de usuario	33
Figura 3.10 Estructura del frame de capa física	35



Figura 3.11 Diagrama del Scrambler	40
Figura 3.12 Diagrama del Codificador Convolutacional	41
Figura 3.13 Puncturing para una tasa de 2/3	41
Figura 3.14 Puncturing para una tasa de 3/4	41
Figura 3.15 Constelaciones de QPSK y 16-QAM	43
Figura 4.1 Diagrama a bloques de los componentes del USRP	46
Figura 4.2 GNU Radio Companion	50
Figura 4.3 LabVIEW con bloques de USRP	51
Figura 4.4 Matlab con bloques de USRP	51
Figura 5.1 Datos iniciales en el diagrama a bloques de LabVIEW	55
Figura 5.2 Datos iniciales en el panel frontal de LabVIEW	55
Figura 5.3 Bloque de CRC-32 en LabVIEW	56
Figura 5.4 Primera etapa del CRC-32	56
Figura 5.5 Segunda etapa del CRC-32	56
Figura 5.6 Selector de Modulación	58
Figura 5.7 Etapa de Padding	59
Figura 5.8 Generación del bit de paridad	59
Figura 5.9 Bloque del Scrambler en LabVIEW	60
Figura 5.10 Diagrama a bloques del Scrambler en LabVIEW	60
Figura 5.11 Codificador Convolutacional de LabVIEW	61
Figura 5.12 Bloque String_to_Bin de LabVIEW	61
Figura 5.13 Diagrama a Bloques del bloque String_to_Bin	62



Figura 5.14 Bloque Puncturing de LabVIEW	62
Figura 5.15 Diagrama a bloques del bloque Puncturing	63
Figura 5.16 Bloque Interleaver de LabVIEW	63
Figura 5.17 Diagrama a bloques del bloque Interleaver	64
Figura 5.18 Bloque ModData de LabVIEW	64
Figura 5.19 Bloque MT Bit to Symbols VI de LabVIEW	65
Figura 5.20 Bloque ModSignal de LabVIEW	65
Figura 5.21 Diagrama a bloques del bloque ModData	65
Figura 5.22 Diagrama a bloques del bloque ModSignal	66
Figura 5.23 Bloque InsertPilots de LabVIEW	66
Figura 5.24 Bloque InsertPilotSignal de LabVIEW	66
Figura 5.25 Diagrama a bloques del bloque InsertPilots	67
Figura 5.26 Diagrama a bloques del bloque InsertPilotSignal	68
Figura 5.27 Bloque Inverse FFT VI de LabVIEW	69
Figura 5.28 Diagrama a bloques de la IFFT para los Datos y para la Señal	69
Figura 5.29 Bloque CP de LabVIEW	70
Figura 5.30 Bloque InsertPreamble de LabVIEW	70
Figura 5.31 Diagrama a bloques del bloque InsertPreamble	70
Figura 5.32 Diagrama a bloques del bloque InsertPreamble (Continuación)	71
Figura 5.33 Diagrama a bloques de inicio de conexión con el USRP e inicio de generación de archivo	72



Figura 5.34 Diagrama a bloques de conexión con el USRP y generación de archivo	72
Figura 5.35 Diagrama de detección de frame	73
Figura 5.36 Bloque ShortSeq Detection2 de LabVIEW	74
Figura 5.37 Diagrama a bloques de corrimiento de ventana en la detección corta	74
Figura 5.38 Diagrama a bloques de correlación en la detección corta	74
Figura 6.1 Ettus Research USRP N210	78
Figura 6.2 Tarjeta hija SBX de 400 MHz a 4400 MHz	79
Figura 6.3 Antena LP0965 de 850 MHz a 6.5 GHz	79
Figura 6.4 Antena VERT2450 de 2.4 GHz a 2.48 GHz	80
Figura 6.5 Espectro con frecuencia central en 2.4 GHz	80
Figura 6.6 Espectro con frecuencia central en 2.48 GHz	80
Figura 6.7 Espectro con frecuencia central en 2.6 GHz	81
Figura 6.8 Escenario para la prueba de sincronía y constelaciones	81
Figura 6.9 Correlación obtenida en la etapa de Sincronización	82
Figura 6.10 Indicadores que muestran la detección de un frame	82
Figura 6.11 Constelación QPSK 1/2 transmitida	83
Figura 6.12 Constelación 64-QAM 1/2 transmitida	83
Figura 6.13 Constelación QPSK 1/2 recibida	84
Figura 6.14 Constelación 64-QAM 1/2 recibida	84
Figura 6.15 Espectro de la señal transmitida	85
Figura 6.16 Espectro de la señal transmitida con multiplicador	86



Figura 6.17 SNR vs BER para QPSK 1/2	89
Figura 6.18 SNR vs BER para 64-QAM 1/2	89
Figura 6.19 Escenario para la prueba de distancia en interiores	90
Figura 6.20 Distancia vs BER para la antena LP0965 en interiores	93
Figura 6.21 Distancia vs BER para la antena VERT2450 en interiores	93
Figura 6.22 Escenario para la prueba de distancia en exteriores	94
Figura 6.23 Distancia vs BER para la antena LP0965 en exteriores	96





Índice de Tablas

Tabla 3.1 Duración en μ s de los IFS	30
Tabla 3.2 Tabla de direcciones MAC	33
Tabla 3.3 Contenido del campo de tasa	36
Tabla 3.4 Valores Dependientes de la modulación	38
Tabla 3.5 Parámetros de tiempo en el frame de capa Física	39
Tabla 3.6 Valores de normalización de las constelaciones	43
Tabla 4.1 Velocidades de procesamiento de las Motherboards de los USRP	48
Tabla 4.2 Modelos de Daughterboards para los USRP	49
Tabla 6.1 Equipo de cómputo utilizado para el transmisor	76
Tabla 6.2 Equipo de cómputo utilizado para el receptor	77
Tabla 6.3 Características Ettus Research USRP N210	78
Tabla 6.4 SNR vs BER medida para QPSK 1/2	88
Tabla 6.5 SNR vs BER medida para 64-QAM 1/2	88
Tabla 6.5 Distancia vs BER para QPSK 1/2 utilizando la antena LP0965 en interiores	91
Tabla 6.6 Distancia vs BER para QPSK 1/2 utilizando la antena VERT2450 en interiores	91
Tabla 6.7 Distancia vs BER para 64-QAM 1/2 utilizando la antena LP0965 en interiores	92
Tabla 6.8 Distancia vs BER para 64-QAM 1/2 utilizando la antena VERT2450 en interiores	92



Tabla 6.9 Distancia vs BER para QPSK 1/2 utilizando la antena LP0965 en exteriores	95
Tabla 6.10 Distancia vs BER para 64-QAM 1/2 utilizando la antena LP0965 en exteriores	95





Capítulo 1

Introducción

En este capítulo se podrán encontrar los rasgos generales que describen a la tesis, dando a conocer el surgimiento de la necesidad de realizar la investigación, el alcance esperado y su importancia en el mundo actual. La mención de los temas antes mencionados es de gran importancia ya que otorgaran un panorama general de la investigación a realizar.

1.1 Antecedentes

La necesidad de los humanos de mantenerse en constante comunicación, los ha llevado a crear diferentes tecnologías para que ésta pueda tener un mayor alcance, a una gran velocidad y al menor costo posible, llegando a crear distintas tecnologías inalámbricas como Global System for Mobile Communications (GSM), Long Term Evolution (LTE) y Wireless Fidelity (Wi-Fi), siendo ésta última muy importante para la vida moderna ya que permite crear una conexión a una LAN (Local Area Network) de una manera eficaz, utilizándose en una gran cantidad de dispositivos.

La tecnología Wi-Fi es la marca registrada del estándares IEEE (Institute of Electrical and Electronics Engineers), específicamente el 802.11, el cual define las características que se deben cumplir para que funcionen correctamente los dispositivos que usen Wi-Fi, así como para que exista interoperabilidad entre ellos. Este estándar contiene distintas versiones (a, b, g, p, ad), variando entre éstas, el tipo de aplicación, las tasas de transferencia que van desde los 2 Mb/s hasta mayores a 1 Gb/s, el ancho de banda (10 Mhz, 20 Mhz, 40 Mhz, etc.) y las frecuencias centrales.



En una sociedad donde existen avances tecnológicos constantemente, el estándar IEEE 802.11 tiene que actualizarse periódicamente, para asegurar que la comunicación regida por el estándar, siempre se encuentre en óptimas condiciones. Lamentablemente al existir tantos cambios en tan poco tiempo, genera grandes costos al tener que crear nueva circuitería para los equipos, por esta razón se decidió crear lo que es llamado Software Defined Radio (SDR), con el cual se pueden crear distintos enlaces de radio con un solo equipo el cual es configurable mediante software, lo que da una gran solución al gasto generado por la actualización de equipos.

La construcción de un dispositivo IEEE 802.11 mediante SDR, es un área muy importante de investigación, ya que con esto se pueden crear dispositivos que puedan resistir el avance tecnológico tan solo cambiando la programación de éste.

1.2 Definición del Problema

Actualmente una gran cantidad de dispositivos utilizan Wi-fi, estandarizados por la norma IEEE 802.11 en sus diferentes versiones, esto se debe a que puede adaptarse a prácticamente cualquier ambiente, con dispositivos relativamente baratos.

El problema que existe con los equipos Wi-fi, es que están hechos con circuitos electrónicos, los cuales son difíciles de actualizar, prácticamente dejándolos obsoletos en poco tiempo por la acelerada actualización de la tecnología, provocando que tanto las empresas como los consumidores gasten grandes cantidades de dinero al obtener las más nuevas versiones, siendo ésta una debilidad que tiene la tecnología estandarizada por las normas IEEE 802.11, agregando que en caso de requerir una mayor cantidad de usuarios en un dispositivo, el precio aumenta considerablemente. Por esta razón es de suma importancia otorgar una mayor flexibilidad a los dispositivos Wi-fi.

Una solución se puede lograr mediante SDR, los cuales son equipos que utilizan módulos con Field Programmable Gate Arrays (FPGAs) intercambiables por el usuario para construir sistemas de transmisión funcionales de acuerdo a las necesidades que



se requieran. Otro punto a favor de estos sistemas es la posibilidad de utilizar distintos softwares para su programación (GNU Radio, LabVIEW y Matlab).

El presente proyecto busca aprovechar las cualidades que el SDR otorga, para implementar la capa física, específicamente de la norma IEEE 802.11g en LabVIEW con bloques reconfigurables, utilizando como radios los equipos Ettus USRP N210 y así lograr obtener el mayor throughput y el menor BER (Bit Error Rate) posibles, además de comprobar la capacidad de los equipos para utilizar otras normas como la versión p y su comportamiento ante la movilidad.

1.3 Objetivos y Metas

La principal tarea es diseñar e implementar la capa física de un transmisor con el software LabView para SDR, basado en el estándar IEEE 802.11g. Una vez realizado esto se analizará el comportamiento del transmisor construido, por medio de un receptor que trabaje en la misma norma e implementado en la misma plataforma.

Es muy importante encontrar el ancho de banda máximo, el BER y el throughput que los equipos Ettus USRP N210 pueden soportar trabajando con el programa a realizar en LabVIEW y así encontrar la viabilidad de utilizar SDR para aplicaciones en tiempo real utilizando el estándar dado.

Otro objetivo es analizar algoritmos de sincronización y así escoger e implementar el más adecuado para el receptor que se comunicará con el transmisor, ya que la sincronización más eficiente que se pueda utilizar con LabVIEW y los equipos USRP N210 ayudará a obtener BER menor.

Al cumplir los objetivos antes mencionados se podrá enunciar las capacidades de SDR para trabajar con el estándar IEEE 802.11, el equipo necesario para trabajar con esta tecnología y así ayudar a investigaciones futuras que decidan utilizar SDR en el país.



1.4 Contribución y Relevancia.

El uso de SDR es aplicado actualmente solo en laboratorios, ya que este tipo de tecnologías no lleva suficiente tiempo en el mercado para saber con exactitud cuáles son sus capacidades, debido a esto la información existente sobre implementar normas IEEE 802.11 es muy escasa, por esta razón la investigación a realizar tendrá aportaciones importantes el campo de los radios definidos por software.

El uso de las herramientas propuestas para hacer funcionar los USRP N210, hará que la Universidad Nacional Autónoma de México sea pionera en el uso de LabVIEW para aplicaciones de estándares IEEE802.11.

La realización distintas mediciones como el BER, PER (Packet Error Rate) y el estado estable del sistema, otorgará un registro formal del rendimiento de OFDM con distintas modulaciones (BPSK, QPSK, 16-QAM y 64-QAM), utilizando randomización, codificación e intercalamiento para SDR.

1.5 Estructura de la tesis

Primeramente se escogen los objetivos los cuales quiere alcanzar el proyecto de acuerdo a distintas pruebas para tecnologías inalámbricas, que medirán la confiabilidad del sistema a implementar.

De acuerdo a los USRP N210 con los que se cuentan, se establecerán las características del canal a utilizar, como es frecuencia, ancho de banda, ganancia de transmisión y velocidad de muestreo, ya que los parámetros dependerán de las capacidades de los equipos.

La presente tesis abarcará desde una descripción detallada del estándar a usar, hasta la metodología que se siguió para realizar la implementación, así como las pruebas que confirman el rendimiento del sistema, por lo que la tesis estará dividida en 6 capítulos.



El capítulo 2 hará una breve descripción de la historia de Wi-fi y de su funcionamiento. También abarcará que es LabVIEW y la forma de programar en este tipo de lenguaje y finalmente se incluirán las investigaciones con más relevancia que trabajen con SDR para la norma IEEE 802.11g.

El capítulo 3 hará una descripción detallada de cómo es que trabaja la capa física en el estándar IEEE 802.11g, abarcando sus seis principales procesos scrambling, coding, puncturing, interleaving, modulation y la IFFT (Inverse Fast Fourier Transform).

El capítulo 4 especificará las funciones, características y formas de uso del equipo SDR, es decir el USRP N210, por qué se decidió programarlo con LabVIEW, así como su utilidad en el proyecto de investigación.

En el capítulo 5 se integrará el código completo realizado en LabVIEW que implementa la capa física para un transmisor IEEE 802.11g y la sincronización de un receptor regido por la misma norma, así como los algoritmos matemáticos que se utilicen como base para el código realizado.

Por último el capítulo 6 describirá los resultados de las pruebas que comprueben el rendimiento del sistema como el Modelo propagación, BER BPSK 3/4, BER 16-QAM 3/4 y movilidad, comprobando si los equipos y el código realizado son útiles para aplicaciones en tiempo real en la norma IEEE 802.11g.



Capítulo 2

Marco Teórico

Introducción

En este capítulo se podrá encontrar una descripción general de las redes Wi-fi, comenzando por su historia hasta las versiones principales del estándar IEEE 802.11, así como las características del lenguaje LabVIEW y la conveniencia de usarlo en el proyecto a realizar. Finalmente se dará a conocer las investigaciones actuales que utilicen SDR para implementar el estándar IEEE 802.11 y las tendencias de próximas investigaciones.

2.1 Redes Wi-fi

2.1.1 Historia del estándar IEEE 802.11

En 1997 fue creado el estándar 802.11 por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) como la primer norma de Redes Inalámbricas de Área Local (WLAN). Después de esto, en el año de 1999 fue creada la Wi-fi Alliance, siendo la unión de distintas corporaciones, quienes impulsaron el estándar registrando la marca Wi-fi la cual certificaría la interoperabilidad entre equipos de distintas compañías.

2.1.2 Versiones del estándar IEEE 802.11

Con el tiempo y la evolución de las tecnologías y necesidades de la sociedad el estándar ha tenido que cambiar e ir agregando diferentes versiones, que se acomoden a las exigencias de los múltiples escenarios donde se pueda implementar una WLAN. Algunos ejemplos de las distintas circunstancias donde se requiera la aplicación de



este tipo de redes son interiores, exteriores, lugares con gran cantidad de objetos que puedan causar múltiples reflexiones, etc. Llegando así a crearse más de 16 versiones, aunque las más utilizadas, las cuales son las que utiliza la mayoría de las personas, se describirán a continuación.

IEEE 802.11a

Esta versión [1] fue oficialmente creada en el año de 1999, la cual especifica el uso de la banda de 5 GHz a una velocidad máxima de 54 Mb/s mediante la técnica de OFDM (Orthogonal Frequency Division Multiplexing). El uso de este modo de multiplexión fue elegido ya que presenta una alta eficiencia espectral, eliminando la interferencia entre portadoras al ser ortogonales.

Un punto a destacar de la versión 'a' es que utiliza modulación adaptiva, en un ancho de banda de 20 MHz, cambiando entre BPSK, QPSK, 16-QAM y 64-QAM dependiendo del SNR que se pueda estimar del canal utilizado.

IEEE 802.11b

La versión 'b' [1] del estándar también fue creada en el año de 1999, pero usando la banda de 2.4 GHz con la técnica de DSSS (Direct Sequence Spread Spectrum), la cual otorgaba un código mediante CCK (Complementary Code Keying) por cada dispositivo que estuviera asociado a la red. Lo cual permite que la aplicación de la versión b de la norma fuera más barata pero solo logrando alcanzar hasta 11 Mb/s de tasa de transferencia, en el mismo ancho de banda que la norma a que es de 20 MHz.

IEEE 802.11g

Esta versión del estándar [1], al igual que la 'b', utiliza la banda de 2.4 GHz, con un ancho de banda de 20 MHz, pero con la técnica de OFDM además de DSSS, con modulación adaptiva, logrando tasas de transferencia hasta de 54 Mb/s al igual que la norma 'a', logrando así la interoperabilidad entre las tres normas.

La versión 'g' es una de las más utilizadas actualmente ya que es muy flexible en la compatibilidad con otras normas a tasas de transferencia aceptables.



IEEE 802.11n

Esta es una versión del estándar [1] que aumenta considerablemente la tasa de transmisión, llegando hasta velocidades de 600 Mb/s, gracias a la utilización de MIMO (Multiple Inputs Multiple Outputs). En esta ocasión se incrementó el ancho de banda a 40 MHz, aunque también puede utilizar 20 MHz.

El estándar también utiliza la banda de 2.4 GHz y 5 GHz, provocando que sea retrocompatible con las normas 'a', 'b' y 'g' utilizando simultáneamente las dos bandas.

IEEE 802.11p

La versión 'p' [1] fue creada para el uso automovilístico, la cual tiene prácticamente las mismas características que la versión 'g', solo que la banda que se utiliza es de 5.9 GHz, utilizando anchos de banda de 5 MHz, 10 Hz y 20 MHz, logrando velocidades desde 1.5 Mb/s hasta 54 Mb/s.

Esta norma fue creada para lograr intercambiar información de manera eficiente entre automóviles, por ejemplo que se informe cuando existe un choque o enviar publicidad.

IEEE 802.11ac

Una de las más nuevas versiones del estándar es la ac [1], ésta realiza un uso más riguroso de las cualidades que se adhirieron en la norma 'n', incrementando el arreglo MIMO que era de 4x4, hasta un arreglo 8x8. Otro cambio de relevancia es que los anchos de banda que se utilizan son de 20 MHz, 40 MHz, 80 MHz y 160 MHz, además la máxima modulación que se puede utilizar es 256-QAM, incrementando así la tasa de transferencia hasta 1300 Mb/s.

Ésta versión solo utiliza la banda de 5 GHz, siendo compatible con la norma 'a' y 'n'.



2.2 LabVIEW

2.2.1 Descripción General de LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) es un lenguaje de programación lanzado a la venta en el año de 1986 por la compañía National Instruments. Este software fue creado para permitir a ingenieros de cualquier rama realizaran programas de prueba, diseño, control, entre otros, los cuales imitarán instrumentos de laboratorio de forma virtual (VI).

La programación gráfica es utilizada por ésta plataforma, es decir se conectan bloques en lugar de escribir texto para así crear una aplicación, siguiendo un flujo datos entre nodos de un diagrama a bloques en lugar de líneas de texto.

Una gran ventaja que tiene este lenguaje de programación, es que puede trabajar en conjunto con otros lenguajes (Matlab, C++, etc.). También puede interactuar con distintos tipos de hardware como FPGA, USB y SDR.

2.2.2 Programación en LabVIEW

Cómo se mencionó anteriormente el tipo de programación utilizado en este software es gráfica, es decir se unen distintos bloques con una función específica, para obtener el resultado deseado.

Un programa en este lenguaje se divide en 2 partes, las cuales se describirán a continuación.

Panel frontal (Figura 2.1): Ésta parte contiene la “interfaz de usuario” del programa, es donde se encuentran los cuadros de dialogo, barras deslizables, selectores, indicadores, gráficas, entre más, en donde la persona que esté utilizando el programa puede cambiar los parámetros de entrada y observar los de salida. Técnicamente los elementos con los que se puede interactuar en el panel frontal son llamados controles e indicadores.

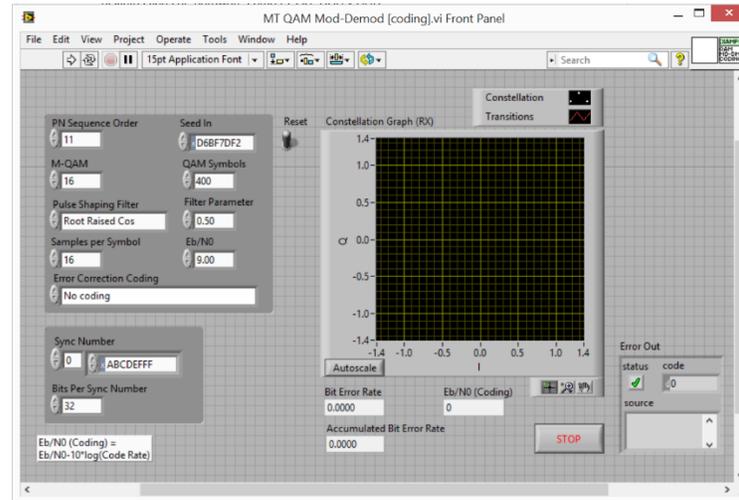


Figura 2.1 – Ejemplo de un panel frontal en LabVIEW.

Diagrama a bloques (Figura 2.2): En esta área es donde se crea el diagrama a bloques del programa a realizar. Se pueden utilizar los bloques incluidos dentro del software como sumas, restas, operaciones entre números complejos hasta codificadores y moduladores, o se pueden programar bloques con otros lenguajes, siendo un ejemplo C++.

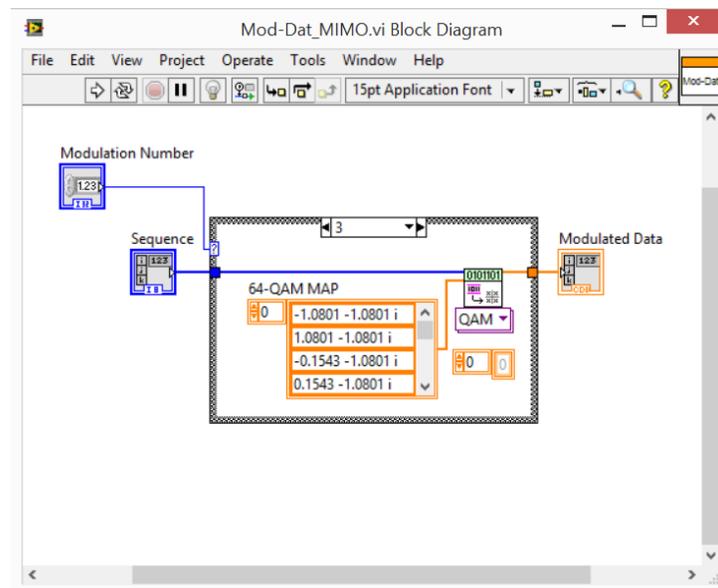


Figura 2.2 – Ejemplo de un diagrama a bloques en LabVIEW.

El panel frontal y el diagrama a bloques interactúan por medio de los llamados controles e indicadores, ya que estos se encuentran en las dos partes. Los controles (Figura 2.3) dan parámetros de entrada en el panel frontal, que son procesados en el diagrama a bloques para así mostrar en el panel frontal los resultados deseados en indicadores (Figura 2.4) que pueden ser cuadros de dialogo, gráficas, luces, etc.



Figura 2.3 – Bloque de un control de un arreglo entero en el panel frontal y en el diagrama a bloques.



Figura 2.4 – Bloque de un indicador de un arreglo complejo en el panel frontal y en el diagrama a bloques.

Para crear un nuevo programa en LabVIEW se necesita crear un nuevo VI, insertando en el diagrama bloques las estructuras necesarias, que se encuentran en la paleta, para programar un algoritmo matemático, el comportamiento de algún instrumento o hasta el control de un dispositivo como el SDR.

En LabVIEW se pueden utilizar distintos tipos de variables (Dato Cadena de Caracteres, Dato Numérico, Dato Booleano, Dato Dinámico, Dato Complejo), los cuales se pueden agrupar en arreglos, matrices o clusters. El tipo de dato a utilizar, es

dado por la estructura usada, la cual tanto en la entrada y en la salida indicará que valores pueden ingresar y que valores se entregarán.

2.2.3 Uso del USRP N210 para transmisión en LabVIEW

A partir de LabVIEW 2013 se incluyeron bloques de control de USRP, entre ellos el N210. Estos bloques comunican los parámetros necesarios, como ancho de banda, velocidad de muestreo, frecuencia central, además entregan los valores a transmitir a las tarjetas del USRP.

La transmisión en un USRP mediante LabVIEW se puede separar en dos partes, inicio de sesión y termino de sesión, las cuales se explicaran a continuación.

Inicio de sesión: En esta sección del programa se inicializa el USRP, para que cree el canal físico con las características deseadas, normalmente se introduce al principio del código para que el USRP reconozca todos los valores que se vayan a transmitir. Los bloques son los siguientes.

- niUSRP Open Tx Session VI (Figura 2.5): Abre la sesión de transmisión para un específico USRP (device name), pasando este parámetro mediante la salida session handle out.

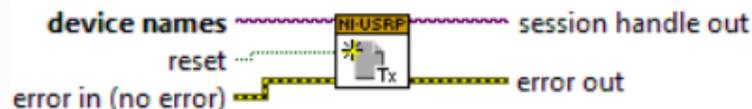


Figura 2.5 - niUSRP Open Tx Session VI.

- niUSRP Configure Signal VI (Figura 2.6): En este bloque se indican todos los parámetros necesarios para crear el medio físico deseado, es decir, el canal o canales a utilizar, el ancho de banda, la frecuencia central, ganancia de transmisión en dB y la antena a utilizar. En la salida entrega mediante session handle out tanto los nuevos parámetros insertados, como los anteriores, además puede mostrar en las salidas coerced, los parámetros reales que indicara al USRP.

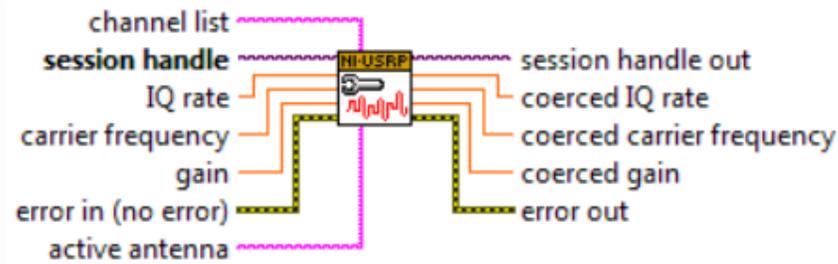


Figura 2.6 - niUSRP Configure Signal VI.

Cierre de sesión: La etapa de cierre de sesión va al final del código ya que se tienen los datos preparados para su transmisión, para que así LabVIEW pueda indicar al USRP los valores exactos que se deseen enviar al aire, Los bloques se explican a continuación.

- niUSRP Write Tx Data (poly) VI (Figura 2.7): Este bloque es el encargado de enviar los datos tal cual los recibe hacia el USRP, del programa creado anteriormente, aceptando valores dobles o complejos, en sí transmite los valores en amplitud de las muestras que se requieran transmitir.

Este bloque recibe las configuraciones otorgadas en la etapa de inicio de sesión mediante la entrada session handle. También se puede retrasar en segundos cada cuando se requiere utilizar el bloque.

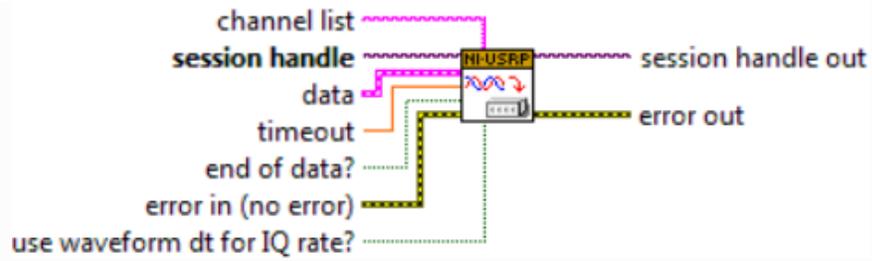


Figura 2.7 - niUSRP Write Tx Data (poly) VI.

- niUSRP Close Session VI (Figura 2.8): Es el bloque final para la transmisión, el cual se encarga de cerrar la sesión abierta anteriormente, para que así termine la transmisión o en su caso reconfigurar el USRP para realizar una nueva transmisión. Lo único que requiere el bloque el device name del dispositivo y así terminar la transmisión.



Figura 2.8 - niUSRP Close Session VI.

Por último cabe mencionar que todos los bloques de USRP tienen una salida y una entrada de error, las cuales se pueden utilizar para un análisis posterior, en caso de que el programa realizado no pueda compilarse apropiadamente.



2.3 Estado del Arte

2.3.1 IEEE 802.11 Wireless LAN Implemented on Software Defined Radio With Hybrid Programmable Architecture

El artículo [2] fue creado en el año 2005 en Japón, el cual describe uno de los primeros intentos en implementar la norma 802.11 mediante SDR. Algo muy relevante en esta publicación es que los autores construyeron todo el dispositivo SDR mediante una Unidad Central de Procesamiento (CPU), cuatro Procesadores Digitales de Señales (DSP) y tres Post Procesadores (PPP) implementados mediante FPGAs. El CPU se encargaba de generar el frame deseado, además de administrar los drivers del protocolo Ethernet, los DSPs modulaban y demodulaban la señal, los PPP se encargaban el demás procesamiento en banda base necesario para transmitir la señal en el aire de acuerdo al estándar 802.11.

Lamentablemente solo se implementaron dos modulaciones, BPSK y QPSK por lo que la velocidad máxima teórica solo era de 2 Mb/s, aunque las pruebas experimentales solo llegaron hasta una velocidad de 1.8 Mb/s. El sistema se hizo para trabajar en una frecuencia de 2.4 GHz y con un ancho de banda de hasta 20 MHz, accedando al medio por CSMA/CA.

Este sistema a pesa de ser uno los primeros en realizarse para el campo de WLAN, logro además de la implementación de una capa física, la implementación de una capa MAC, la cual operaba mediante drivers de Ethernet instalados en el sistema, logrando así obtener un Frame Error Rate (FER), de 10^{-4} .

A pesar de ser un sistema con buenos resultados, al ser construido desde cero, provoca que sea para usos muy exclusivos, de grandes dimensiones y de difícil acceso para la población, por lo que no cualquiera puede trabajar con este equipo y sea



demasiado costoso, además de solo contemplar dos modulaciones de las 8 que hay en el estándar.

2.3.2 Performance evaluation for OFDM Link Based on LabVIEW and USRP

En este artículo [3] principalmente, se describe que es OFDM matemáticamente además de la demostración de la velocidad de muestreo necesaria para poder enviar y recibir información satisfactoriamente a través de medios digitales, en este caso el USRP. También demuestran la creación para implementar un transmisor y un receptor OFDM en LabVIEW, enviando información al aire por medio de un USRP 2, para anchos de banda de 1.5 MHz a 2MHz a una frecuencia de 915 MHz. Es muy importante destacar que la implementación de OFDM se hizo para 256 portadoras con las modulaciones QPSK, 16 QAM y 64 QAM.

Al momento de enviar y recibir información se mostraron las constelaciones de las modulaciones dichas anteriormente, demostrando así la estabilidad de LabVIEW, para mantener una comunicación estable utilizando SDR. Otra gráfica importante que se obtuvo en este proyecto, fue el diagrama de ojo para modulaciones QAM, donde se puede observar que los bits llegan de forma correcta logrando así interpretarlos de manera satisfactoria.

Los autores como conclusión dieron, que efectivamente LabVIEW puede controlar apropiadamente SDR para implementar comunicaciones inalámbricas en interiores, en especial sistemas OFDM, sirviendo los resultados obtenidos para investigaciones futuras.

2.3.3 IEEE 802.11p Transmission Using GNURadio

Este artículo [4] creado en la Universidad de Salento, describe una de las primeras implementaciones de un transmisor de la norma IEEE 802.11, en este caso la versión



'p', utilizando el equipo USRP2, lo cual es de suma importancia, ya que demuestra la eficacia de estos equipos para la aplicación de los estándares IEEE.

El proyecto se realizó en dos pasos, primero se simularon los pasos descritos en el estándar (scrambling, coding, puncturing, interleave, IFFT), para 64 portadoras, verificando así el buen funcionamiento del código que se realizó. Después se continuó con la implementación física, en esta ocasión utilizando GNU Radio, para programar el procesamiento de la información, utilizando USRP2 para transmitir los datos procesados con anterioridad, una frecuencia de 5.9 GHz con una daughterboard instalada en el equipo SDR, llamada XCVR2450, la cual puede trabajar en frecuencias de 2.4-2.5 GHz y 4.9-5.9 GHz. Los anchos de banda utilizados fueron de 10 MHz y 20 MHz.

Finalmente se comparó el espectro enviado por el USRP2 y en equipo comercial (Atheros-based), observando que el USRP generaba picos en ± 6 MHz, los cuales no estaban presentes en el equipo Atheros. Además se graficó el SNR vs FER, obteniendo casi los mismos resultados entre los dos equipos.

La conclusión que se dio fue que el estándar IEEE 802.11 p, a pesar de su complejo procesamiento, puede implementarse de una manera eficiente utilizando SDR.

2.3.4 An IEEE 802.11a/g/p OFDM in Receiver for GNURadio

El artículo [5] describe principalmente la implementación de un receptor para los estándares IEEE 802.11 'a', 'g' y 'p', utilizando como transmisor el realizado en el artículo *IEEE 802.11p Transmission Using GNURadio*, aunque ampliando el ancho de banda hasta 20 MHz, siendo esta investigación la primera en llegar a un ancho de banda tan amplio.

Para lograr obtener los datos transmitidos se agregaron nuevos procesamiento en el receptor además de FFT, Demodulación, Deinterleave, Depuncturing, Decodificación y Descrambling, para obtener una recepción lo más fiable posible (Sincronización,



Alineamiento de Símbolo y Ecuación). Después de esto se comparó la recepción y la transmisión, observando las constelaciones y midiendo el FER, con cinco equipos comerciales que utilizaran 10 MHz y 20 MHz de ancho de banda para los respectivos estándares.

Los resultados obtenidos fueron muy similares entre los equipos comerciales y el de SDR, aunque las constelaciones en este último estaban un poco más expandidas, pero se demostró que se puede implementar un sistema completo de comunicación comercial (Transmisor y Receptor) mediante SDR.

2.3.5 An IEEE 802.11 MAC Software Defined Radio Implementation for Experimental Wireless Communications and Networking Research

En este artículo [6] se realiza la implementación de una capa MAC mediante el uso de SDR, basándose en el transmisor y receptor del artículo *Decoding IEEE 802.11a/g/p OFDM in Software using GNURadio*. Tanto el transmisor, como el receptor se modificaron para que estos pudieran enviar ACKS, NACKS, CTS, RTS y calcular DIFS, para así lograr poder enviar retransmisiones y ocupar solo cuando éste se encuentre vacío.

Para lograr implementar la capa MAC de manera satisfactoria en este caso se tuvieron que contemplar retardos, para que así los dispositivos que intervinieran en la comunicación pudieran entrar en competencia y así ocupar el canal uno a la vez, contando así desde el procesamiento de la PC y el USRP, hasta la duración de los DIFS.

Finalmente se utilizó Wireshark, para captar los frames enviados, confirmando así que estos provenían de los USRP y observando si la información enviada eran ACKS, RETRANSMISIONES, etc. Al momento de realizar la captura de datos se confirmó que efectivamente la información era la transmitida por los USRP y además se logró



reconocer el tipo de frame, concluyendo que a pesar del alto procesamiento que la capa MAC y Física requieren, es viable implantarlas en conjunto.



Capítulo 3

Estándar IEEE 802.11g

Introducción

La norma 'g' es de suma importancia en la vida cotidiana actual, ya que por el ancho de banda que maneja y sus frecuencias de operación la hace compatible con otras versiones del estándar 802.11, facilitando su implementación. Esto se puede observar en las especificaciones de los equipos que utilizan WLAN, ya que casi todas sus tarjetas de red, están basadas en la versión 'g'.

En este capítulo se podrá encontrar la descripción del estándar IEEE 802.11g, explicando la subcapa de Control de Acceso al Medio (MAC) y la capa Física (PHY), para que así exista un amplio entendimiento del funcionamiento del estándar y como es que éste indica la forma de procesar los datos y mandar por medio aéreo información.

3.1 Aspectos Básicos del estándar IEEE 802.11

3.1.1 Topología IEEE 802.11

Dentro del estándar IEEE 802.11 existen aspectos generales que corresponden a todas las versiones existentes. Tal como la topología y los elementos que pertenecen a redes WLAN dentro de la norma, como se muestra en la figura 3.1.

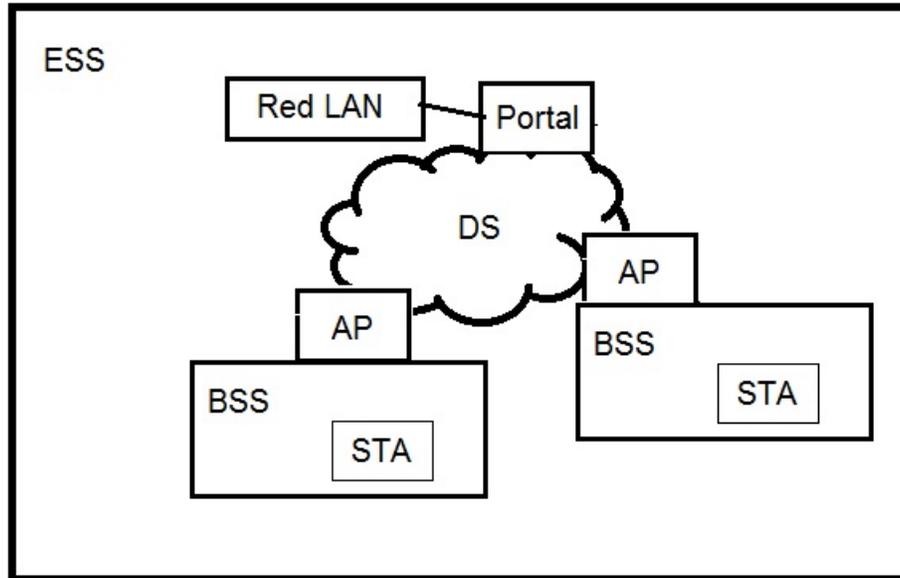


Figura 3.1 – Topología de la norma IEEE 802.11.

Donde la entidad más grande es llamado Conjunto de Servicios Extendidos (ESS), el cual se encuentra conformado por dos o más Conjuntos de Servicios Básicos (BSS) conectados a un Sistema de Distribución (DS) (por lo general la troncal de una LAN) mediante un Access Point (AP). Para que exista una conexión entre redes WLAN y redes LAN se utiliza un Portal.

Los dispositivos básicos de la red 802.11, son llamados estaciones (STA), los cuales son los equipos que utiliza un usuario común para acceder a redes de éste tipo.

Los BBS al no tener relación con otros BBS se les llama Conjunto Independiente de Servicios Básicos (IBBS), tal como n redes tipo Ad-Hoc.



3.1.2 Esquemas de transmisión

Para el estándar IEEE 802.11 existen distintas formas de transmitir la señal, las cuales se explican a continuación.

Direct Sequence Spread Spectrum (DSSS): Es una técnica la cual reparte la potencia en el ancho de banda disponible relacionando una secuencia pseudo-aleatoria '11' con cada bit de datos '1' o '0'.

Utiliza la frecuencia de 2.4 GHz, con 3 canales disponibles en un ancho de banda de 5 MHz, alcanzando velocidades de 1 Mb/s con modulación DBPSK y 2 Mb/s con modulación DQPSK.

Frequency Hopping Spread Spectrum (FHSS): Es una técnica envía ráfagas de información cambiando la frecuencia basándose en una secuencia pseudo-aleatoria.

Utiliza la frecuencia de 2.4 GHz, con 79 canales disponibles en un ancho de banda de 1 MHz, alcanzando velocidades de 1 Mb/s con modulación DBPSK y 2 Mb/s con modulación DQPSK.

Infrared Light (IR): Es parte del estándar que describe la forma de comunicación mediante pulsos infrarrojos, utilizando una modulación 16-PPM (Pulse Position Modulation), la cual varía la posición del pulso para representar los bits, dejando la amplitud y el ancho fijos. Cada 4 bits son relacionados con 16 símbolos PPM.

Orthogonal Frequency Division Multiplexing (OFDM): Fue incluida hasta la versión 'a' del estándar debido a la creciente demanda de usuarios en las WLAN.

Es un esquema multiportadora, lo que provoca que el sistema sea selectivo en frecuencia, donde el ruido o interferencias afectan solo un segmento de portadoras, contrario a esquemas de una sola portadora que afecta todo el sistema. Las portadoras son ortogonales una de otra, eliminando la interferencia entre portadoras adyacentes, conformando un símbolo OFDM. Gracias a esto se pueden condensar más portadoras en un ancho de banda generando una gran eficiencia espectral en el sistema (Figura 3.2). En contraste FDM que no es ortogonal necesita bandas de guarda para impedir

la interferencia entre portadoras adyacentes, desperdiciándose ancho de banda (Figura 3.2).

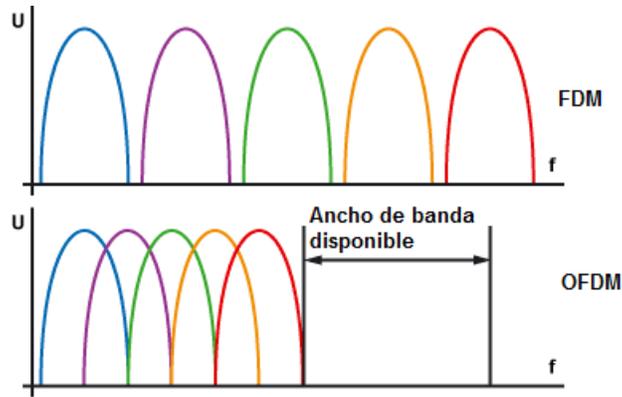


Figura 3.2 – Comparación de uso de ancho de banda de FDM contra OFDM.

Cuando una señal viaja en el aire y choca contra objetos, la señal se divide en un frente de onda, generando distintas copias de la señal pero de menos potencia. Estas señales al seguir caminos diferentes hacia el receptor, pueden adelantarse o retrasarse respecto a la señal original y señales posteriores, provocando una Interferencia Intersimbólica, degradando o generando errores en las señales que llegan al receptor.

Para eliminar la Interferencia Intersimbólica se agrega un Ciclo Prefijo (CP), que extiende periódicamente la señal, evitando que los símbolos se traslapen.

Para generar un símbolo OFDM, se toman los datos de forma serial antes de ser modulados, se convierten a paralelo, utilizando un convertidor Serial-Paralelo, donde las salidas del convertidor corresponden al número de portadoras del símbolo OFDM, normalmente siendo números complejos ya que es una etapa posterior a la modulación. Finalmente se aplica la Transformada inversa de Fourier (IFFT) y se aplica un Prefijo Cíclico (1/4 de la señal para el estándar IEEE 802.11). El esquema de un transmisor OFDM se puede ver en la figura 3.3.

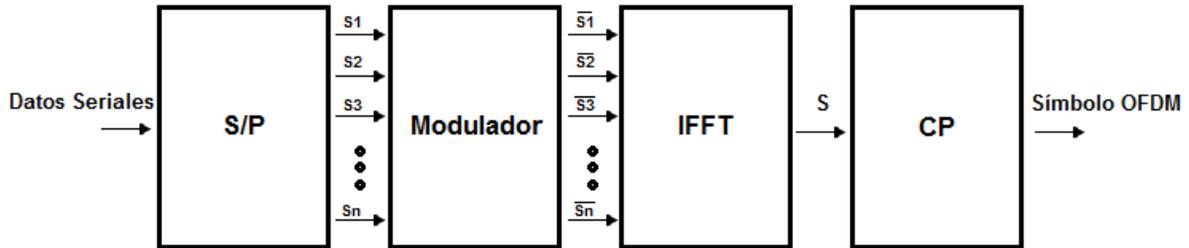


Figura 3.3 – Esquema de un Transmisor OFDM.

Para la etapa de recepción se hace el proceso inverso como se observa en la figura 3.4.

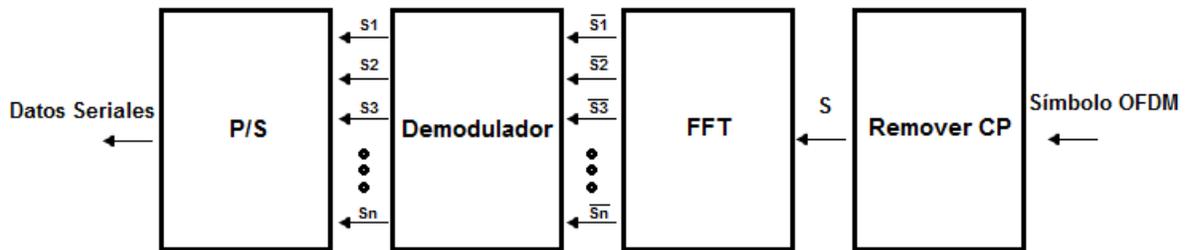


Figura 3.4 – Esquema de un Receptor OFDM.

Para el estándar IEEE 802.11g se utilizan las frecuencias de 2.4 GHz y 5.9 GHz, con 64 portadoras y 13 canales disponibles en un ancho de banda de 20 MHz, alcanzando velocidades de 6 Mb/s, 9 Mb/s, 12 Mb/s, 18 Mb/s, 24 Mb/s, 36 Mb/s, 48 Mb/s, 54 Mb/s, con modulaciones BPSK 1/2, BPSK 3/4, QPSK 1/2, QPSK 3/4, 16-QAM 1/2, 16-QAM 3/4, 64-QAM 2/3 y 64-QAM 3/4 respectivamente.

3.2 Subcapa MAC

3.2.1 Carrier Sense Multiple Access With Collision Avoidance (CSMA/CA)

El protocolo CSMA/CA es usado para minimizar las colisiones al momento de realizar la transmisión de datos por el medio aéreo, el cual fue necesario implementar ya que al existir n número de transmisiones en el aire, es imposible escuchar el canal como en Carrier Sense Multiple Access with Collision Detection (CSMA/CD). En el estándar 802.11g se utilizan dos tipos de censado el físico y el virtual.

El censado físico de acuerdo con Tanenbaum en su libro “Computer Networks” [7], CSMA/CA utiliza el algoritmo de exponential backoff después de una colisión, censando el canal después de esperar un tiempo aleatorio que va de 0 a 15, buscando que no exista ninguna señal por un corto periodo de tiempo (Random Backoff), así se hace una cuenta regresiva de slots ideales, pausando el envío de frames, cuando se alcanza el 0 se envía un frame, si éste llega a su destino el receptor inmediatamente envía un ACK, si el transmisor no recibe la confirmación, se toma como una colisión y se dobla el periodo de backoff, intentándose los mismos pasos de nuevo.

En la figura 3.1 se puede observar que cuando la estación A se encuentra enviando un frame, las estaciones B y C están listas para transmitir, por lo que esperan a que el canal sea ideal y así comenzar su periodo aleatorio de backoff, logrando así que C y B logren transmitir después de A evitando colisiones.

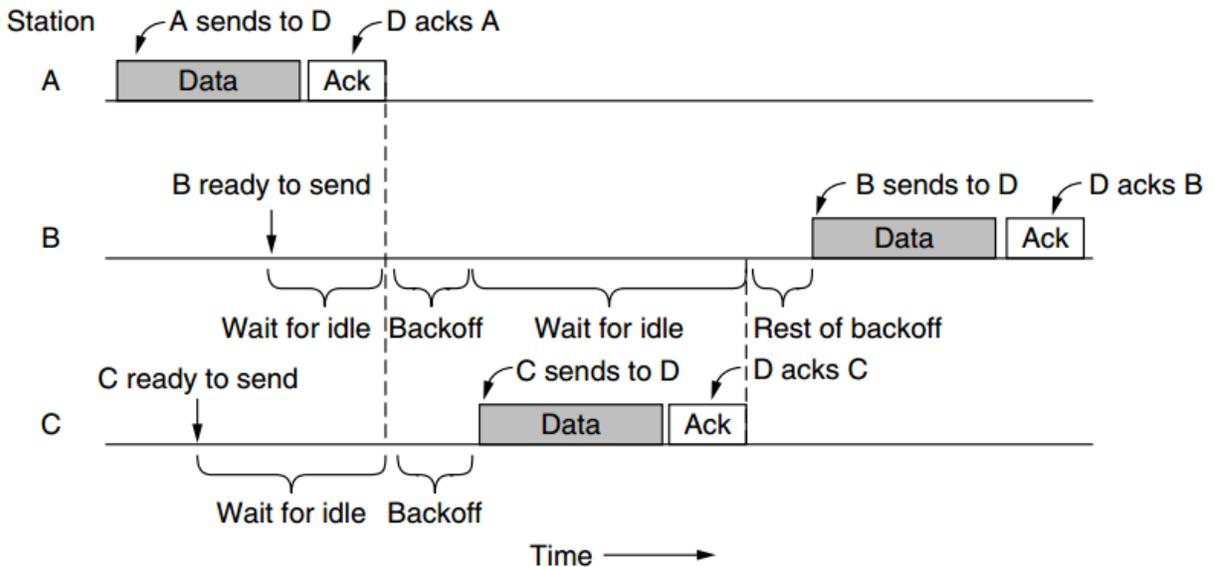


Figura 3.5 – Envío de un frame con CSMA/CA.

3.2.2 Distributed Coordination Function (DCF)

Al momento de realizar una transmisión en una WLAN además de no poder escuchar el canal, existe otro problema el cual es llamado “estaciones ocultas” (Figura 3.6), el cual consiste en que si una estación C desea transmitir a una estación B pero una estación A que se encuentra transmitiendo, la cual está en el rango de B pero no en el rango de C, se provoca una colisión por lo que la comunicación falla.

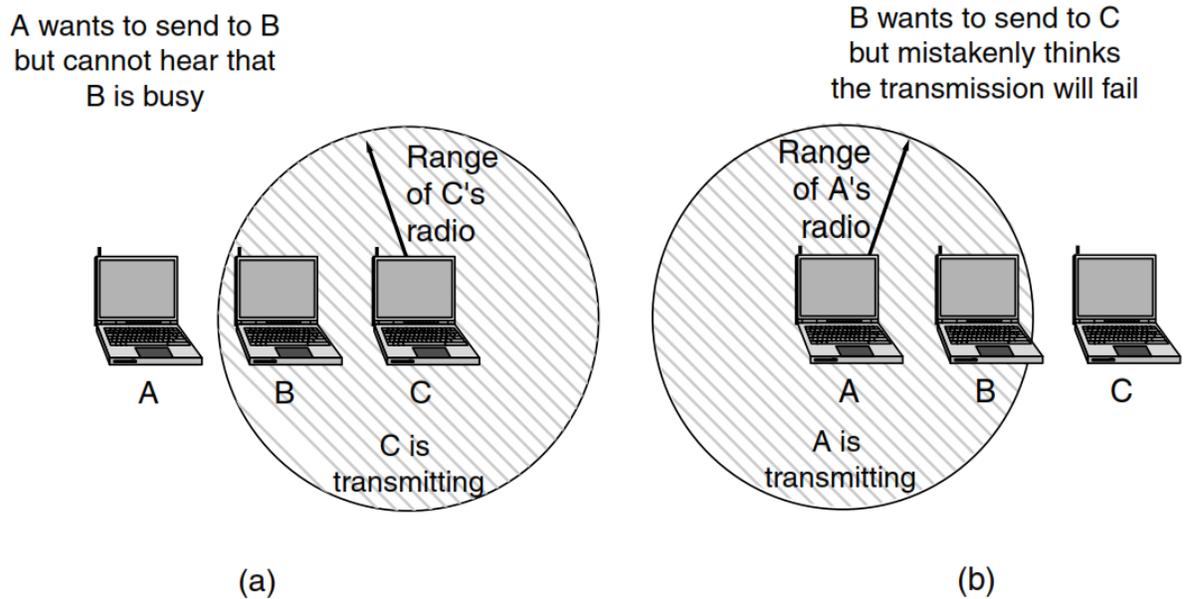


Figura 3.6 – Estación A oculta de C.

De acuerdo con el estándar 802.11 [1], para solucionar el problema de estaciones ocultas se utiliza además de un censado físico, un censado virtual, el cual consiste en enviar un Network Allocation Vector (NAV), enviándose inmediatamente después del periodo de backoff, indicando cuanto tiempo el canal será ocupado, para que así las estaciones cercanas no transmitan o reciban durante ese intervalo.

Para complementar el mecanismo de censado virtual se utilizan frames llamados Request To Send (RTS) y Clear To Send (CTS), logrando que dos o más estaciones no transmitan al mismo tiempo y así evitar colisiones. El mecanismo de RTS/CTS (Figura 3.7) consiste en enviar un RTS después de la transmisión del NAV, pidiendo al receptor responda con un CTS. En caso de que el transmisor nunca reciba la respuesta de que el canal está libre se interpreta como una colisión o que el canal está ocupado, regresando al periodo de backoff para intentar la transmisión más tarde, si se recibe el CTS se envía el frame, evitando que otras estaciones transmitan en ese momento.

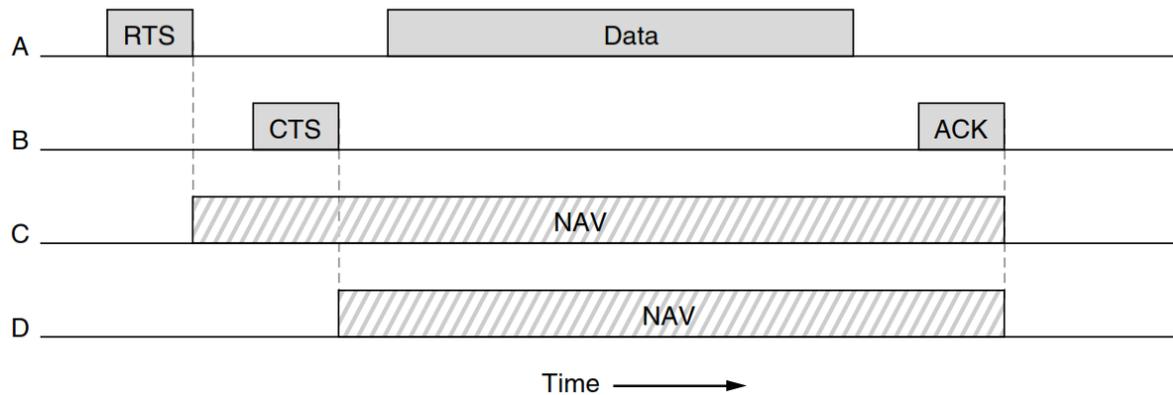


Figura 3.7 – Censado virtual utilizando RTS/CTS.

El problema de utilizar un mecanismo como éste, es que al momento de enviar frames muy pequeños, puede suceder que los RTS, CTS y ACKs sean más grandes que el total de bits datos enviados, provocando que la fiabilidad de la red sea menor, ya que se reduce el throughput total del enlace.

Es muy importante mencionar que al momento de enviar mensajes de broadcast no se puede utilizar RTS/CTS, ya que se recibirían múltiples CTS, los cuales interferirían con la comunicación, por lo que en estos casos se envían los mensajes sin esperar confirmación alguna.

Existen otras dos formas de prevenir las colisiones en la norma 802.11g. Point Coordination Function (PCF), donde un AP se encarga de regular las transmisiones iniciando periodos de backoff entre cada beacon. Hybrid Coordination Function (HCF) es muy parecido a PCF pero este inicia los periodos de backoff en cualquier momento.

Los tiempos donde el canal permanece en estado idle o Interframe Space (IFS), es decir donde está libre para transmitir se mide de diferentes formas dependiendo del momento en que se esté transmitiendo y del método de prevención de colisiones, dividiéndose en cinco tipos.



Short Interframe Space (SIFS): Es el más corto de los IFS se envía previo a los ACK, CTS y entre cada MAC Protocol Data Unit (MPDU) de una ráfaga de fragmentos, para así compensar el retraso que pueda generarse durante el proceso recibió información antes de que ésta fuera enviada.

PCF Interframe Space (PIFS): Es usado solamente para PCF y es más largo que el SIFS pero más corto que el DIFS, lo cual sirve para activar el AP, que funciona como gestor de las transmisiones, antes de que las estaciones entren al periodo de backoff.

DCF Interframe Space (DIFS): Es el periodo utilizado en DCF para saber si el canal está siendo ocupado y así saber si se puede realizar una comunicación o entrar a un periodo de backoff.

Arbitrary Interframe Space (AIFS): Es el tiempo que se da entre servicios para priorizar el envío de un tipo de datos sobre otro, es decir es el tiempo que se otorga para que exista una Calidad de Servicio (QoS).

Extended Interframe Space (EIFS): Es la extensión de tiempo que se da en las estaciones que no lograron decodificar un frame, ya que el campo FCS fue incorrecto, para que así la estación que si recibió el mensaje correctamente pueda mandar un ACK libre de interferencias para la estación transmisora.

A pesar de que los IFS están incluidos en todas las versiones de la norma IEEE 802.11, los tiempos de duración de cada uno de los IFS varía. Para la versión 'g' la duración se puede observar en la tabla 3.1, además su forma de utilizarse en el tiempo se encuentra en la Figura 3.4.

IFS	Duración (µs)
SIFS	10
PIFS	9 o 20
DIFS	9 o 20
AIFS	EIFS- DIFS+AIFS[AC]
EIFS	SIFS+DIFS+Tiempo de ACK

Tabla 3.1 – Duración en µs de los IFS.

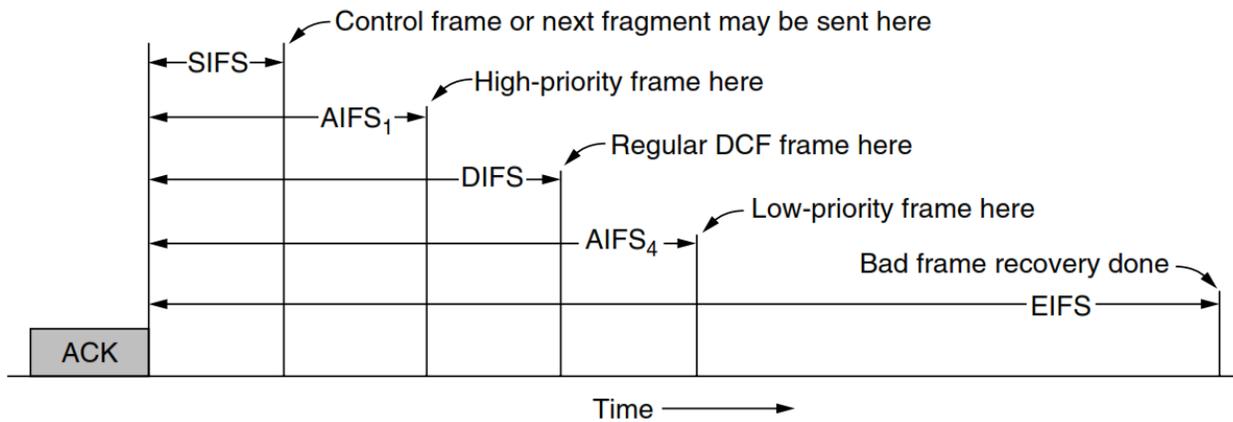


Figura 3.8 – Representación en el tiempo de los IFS del estándar IEEE 802.11g.



3.2.3 Tipos de frame

En la norma IEEE 802.11 [1] existen distintos tipos de frame que son enviados para una acción en específico como un ACK, un RTS, etc. Para que así se pueda realizar la comunicación correctamente, los frames se explicaran a continuación.

Frame de ACK: Es el tipo de frame que se envía inmediatamente antes de los frames de datos, administración, BlockAckReq control, BlockAck control, o PS-Poll control frame. Se utiliza para confirmar la llegada de frames.

Frame de PS-Poll: Es enviado por los Acces Point (AP), para decirle a los dispositivos registrados que no estén transmitiendo, que entren a modo de ahorro de energía, donde el campo de AID indica en cuanto tiempo los dispositivo deberán despertar.

Frame de Contention Free-End (CF-End): Es Frame enviado para indicar el periodo Libre de contención en el método de acceso PCF.

Frame de Contention Free-End+Contention Free-ACK (CF-End+CF-ACK): Es Frame enviado para indicar el periodo Libre de contención con recepción de confirmación en el método de acceso PCF.

Frame de Block ACK Request (BAR): Es utilizado para pedir el bloqueo de ACK, ya que se requiere utilizar QoS.

Frame de Block ACK (BlockACK): Es usado como confirmación por el bloqueo de QoS en lugar de confirmar cada frame independientemente.

Frame de Administración (Management): Es utilizado para activar las estaciones que se encuentran en modo de ahorro de energía, para así establecer una comunicación.

Frame de Beacon: Es enviado periódicamente para anunciar la persistencia y la información relevante de un AP a las NICs que se encuentren dentro de su alcance.

Frame de IBSS ATIM: Es en indicador del Mapa de tráfico del IBSS.



Frame de Disociación: Frame enviado por una estación para disociarse de un AP.

Frame de petición de Asociamiento: Frame enviado por una estación para asociarse a un AP.

Frame de respuesta de Asociamiento: Frame enviado por un AP, con la información necesaria para que una estación pueda asociarse.

Frame de petición de Reasociamiento: Frame enviado por una estación para reasociarse a un AP.

Frame de respuesta de Reasociamiento: Frame enviado por un AP, con la información necesaria para que una estación pueda reasociarse.

Frame de petición de Prueba: Una estación envía este frame para solicitar el escaneo del área existente en el AP.

Frame de respuesta de Prueba: Envía los parámetros obtenido en el escaneo del área.

Frame de Autenticación: Envía los parámetros necesarios para autenticar una estación en un AP.

Frame de Deautenticación: Envía los parámetros necesarios para deautenticar una estación en un AP.

Frame de Acción: Utilizado para disparar una acción en una célula.

Frame de Datos: Es el tipo de frame utilizado para enviar los datos del usuario, es el más complejo de todos los frames, ya que tiene los distintos campos que se utilizan para que exista la mejor transmisión y recepción posible. En la figura 3.5 se puede observar la estructura del frame.

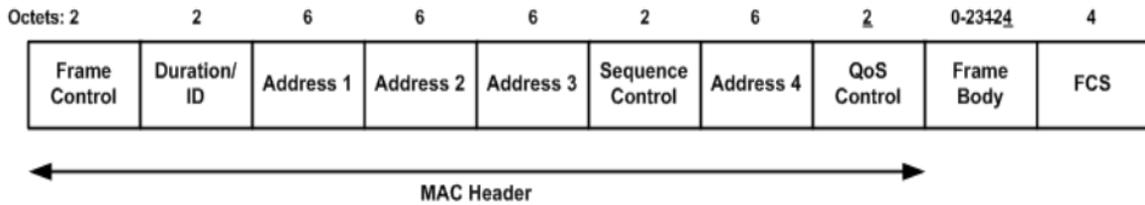


Figura 3.9 – Estructura del frame de datos de usuario.

Los distintos campos del frame de datos se explicarán a continuación.

Frame control: Consta de dos bytes los cuales indican que direcciones serán incluidas en el frame, en la tabla 2 se puede observar las combinaciones existentes para este campo.

Duration/ID: Para frames de control transmite el identificador de asociación y para los frames non-QoS se usa el valor constante 32768.

Address 1: Consta de seis bytes y contiene la dirección MAC de destino (Tabla 3.2).

Address 2: Consta de seis bytes y contiene la dirección MAC de la fuente (Tabla 3.2).

Address 3: Consta de seis bytes y contiene normalmente la dirección MAC BSSID (Tabla 2).

Address 4: Consta de seis bytes y normalmente está vacía (Tabla 3.2).

To DS	From DS	Address 1	Address 2	Address 3	Address 4
0	0	RA = DA	TA = SA	BSSID	N/A
0	1	RA = DA	TA = BSSID	SA	N/A
1	0	RA = BSSID	TA = SA	DA	N/A
1	1	RA	TA	DA	SA

Tabla 3.2 – Tabla de direcciones MAC.



Donde

- DA es el campo de destino.
- SA es el campo de fuente.
- RA es el campo de la dirección multicast o broadcast en el receptor.
- TA es el campo de la dirección multicast o broadcast en el transmisor.

Sequence Control: Es el campo de control de secuencia para QoS.

QoS Control: Si el valor es 1 la información será transmitida como frame de datos QoS.

Frame Body: Contiene los datos de usuario.

Frame Check Sequence: Contiene secuencia obtenida por el algoritmo de CRC-32, para la verificación de la recepción correcta de la información.

El polinomio utilizado es:

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 \quad (3-1)$$

3.3 Capa Física (PHY)

La capa física del estándar IEEE 802.11g, se encarga de indicar los pasos necesarios para que exista un procesamiento adecuado de los datos para que puedan ser transmitidos correctamente y así llegar al receptor minimizando los errores.

Ya que OFDM es eficiente para lidiar con multitrayectorias, robusto contra interferencias de banda angosta y tiene una alta eficiencia espectral, se eligió para el proyecto propuesto, por lo que capa física que se explicará a continuación será para este caso en particular.

3.3.1 Descripción del Frame de Capa Física

En esta capa se agregan nuevos encabezados los cuales son llamados PHY Layer Convergence Procedure (PLCP), generando un nuevo frame (Figura 3.6) el cual está dividido en tres partes, símbolos de preámbulo PLCP, señal OFDM y Datos.

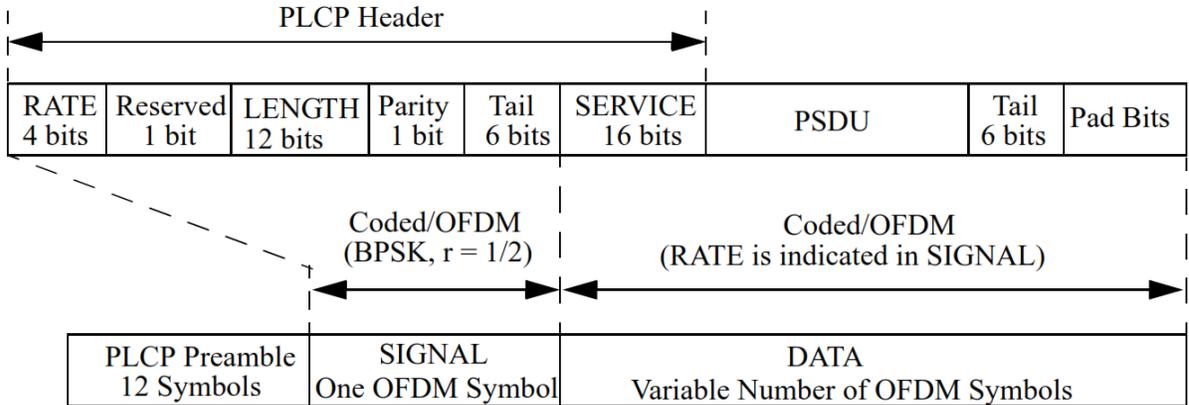


Figura 3.10 – Estructura del frame de capa física.

Los campos del frame se explicarán a continuación.

Símbolos de Preámbulo PLCP: Consta de una secuencia de entrenamiento larga (3-2) y una secuencia de entrenamiento corta (3-3). Se agregan al final del frame, con una duración de 16 μ s y se utilizan en la recepción para la sincronización, detección de símbolo y la ecualización de canal.

$$L_{-26,26} = \{1,1, -1, -1,1,1, -1,1, -1,1,1,1,1,1,1, -1, -1,1,1, -1,1, -1,1,1,1,1,0,1, -1, -1,1,1, -1,1, -1,1, -1, -1, -1, -1, -1, -1, -1,1,1, -1, -1,1, -1,1, -1,1,1,1,1\}$$

(3-2)

$$S_{-26,26} = \sqrt{13/6}\{0,0,1 + j, 0,0,0, -1 - j, 0,0,0, 1 + j, 0,0,0, -1 - j, 0,0,0, -1 - j, 0,0,0, 1 + j, 0,0,0, 0,0,0, -1 - j, 0,0,0, -1 - j, 0,0,0, 1 + j, 0,0,0, 1 + j, 0,0,0, 1 + j, 0,0,0, 1 + j, 0,0\}$$

(3-3)



Tasa: Consta de cuatro bits y es el campo que indica la modulación que se utiliza en los datos (Tabla 3.3).

R1-R4	Tasa (Mb/s)(Ancho de banda de 20 MHz)	Tasa (Mb/s)(Ancho de banda de 20 MHz)
1101	6	3
1111	9	4.5
0101	12	6
0111	18	9
1001	24	12
1011	36	18
0001	48	24
0011	54	27

Tabla 3.3 – Contenido del campo de tasa.

Reservado: Es un bit reservado para un futuro uso.

Longitud: Consta de 12 bits el cual indica el número de octetos del PLCP Service Data Unit (PSDU) de capa MAC que es pasado a la capa PHY.

Paridad: Campo utilizado para agregar un bit de paridad.

Cola (Señal): Son seis bits en '0' que funcionan como cola para la parte de Señal del frame.

Servicio: Campo de 16 bits donde los primeros 6 bits son '0' y los bits restantes son reservados para futuro uso, siendo estos '0'.

PSDU: Es un espacio de longitud variable. Son los datos provenientes de la capa MAC hacia la capa PHY.



Cola (Datos): Son seis bits en '0' que funcionan como cola para la parte de Señal del frame.

Pad Bits: Son bits que se agregan para que los bits de datos, de los símbolos OFDM, sean múltiplo de N_{DBPS} (Tabla 3.4), donde el campo de datos del frame debe ser múltiplo de N_{CBPS} (Tabla 3.4). Al final de los bits de relleno se agregan 6 bits en '0' para completar los octetos juntos con los bits de cola. El número de símbolos OFDM (N_{SYM}), número de bits en el campo de Datos (N_{DATA}) y el número de Pad bits son calculados de acuerdo a la longitud del mensaje PSDU de capa MAC, como se explica a continuación.

$$N_{SYM} = Ceiling((16 + 8 \times LENGTH + 6)/N_{DBPS}) \quad (3-4)$$

$$N_{DATA} = N_{SYM} \times N_{DBPS} \quad (3-5)$$

$$N_{PAD} = N_{DATA} - (16 + 8 \times LENGTH + 6) \quad (3-6)$$

Finalmente se agrega una cantidad ceros igual a N_{PAD} , para que así el mensaje este completo.



Modulación	Tasa de Codificación (R)	Bits codificados por subportadora (N_{BPSC})	Bits codificados por símbolo OFDM (N_{CBPS})	Tasa de datos por símbolo OFDM (N_{DBPS})	Tasa (Mb/s)(Ancho de banda de 20 MHz)	Tasa (Mb/s)(Ancho de banda de 20 MHz)
BPSK	1/2	1	48	24	6	3
BPSK	3/4	1	48	36	9	4.5
QPSK	1/2	2	96	48	12	6
QPSK	3/4	2	96	72	18	9
16-QAM	1/2	4	192	96	24	12
16-QAM	3/4	4	192	144	36	18
64-QAM	2/3	6	288	192	48	24
64-QAM	3/4	6	288	216	54	27

Tabla 3.4 – Valores Dependientes de la modulación.

En algunos campos también se agregan prefijos cíclicos para permitir que la información llegue a tiempo al receptor. Los tiempos finales del frame se muestran en la tabla 3.5.

Parámetro	Valor para un ancho de banda de 20 MHz	Valor para un ancho de banda de 10 MHz
Número de subportadoras de datos (N_{SD})	48	48
Número de subportadoras piloto (N_{SP})	4	4
Número total de subportadoras (N_{ST})	$52(N_{SD} + N_{SP})$	$52(N_{SD} + N_{SP})$
Espaciamiento en frecuencia de las subportadoras (ΔF)	20 MHz/64	10 MHz/64
Transformada Rápida de Fourier (T_{FFT})	$1/\Delta F$	$1/\Delta F$
Duración del Preámbulo ($T_{PREAMBLE}$)	$16\mu S (T_{SHORT} + T_{LONG})$	$32\mu S (T_{SHORT} + T_{LONG})$
Duración del campo de Señal (T_{SIGNAL})	$4\mu S (T_{GI} + T_{FFT})$	$8\mu S (T_{GI} + T_{FFT})$
Duración del ciclo prefijo (T_{GI})	$0.8\mu S (T_{FFT}/4)$	$1.6\mu S (T_{FFT}/4)$
Duración del ciclo prefijo de la secuencia de entrenamiento (T_{GI2})	$1.6\mu S (T_{FFT}/2)$	$3.2\mu S (T_{FFT}/2)$
Intervalo de símbolo (T_{SYM})	$4\mu S (T_{GI} + T_{FFT})$	$8\mu S (T_{GI} + T_{FFT})$
Duración de la secuencia de entrenamiento corta (T_{SHORT})	$8\mu S (10 \times T_{FFT}/4)$	$16\mu S (10 \times T_{FFT}/4)$
Duración de la secuencia de entrenamiento larga (T_{LONG})	$8\mu S (T_{GI2} + 2 \times T_{FFT})$	$16\mu S (T_{GI2} + 2 \times T_{FFT})$

Tabla 3.5 – Parámetros de tiempo en el frame de capa Física.

3.3.2 Procesamiento de los Datos

Para que la información pueda llegar lo más íntegra posible al receptor, en el estándar IEEE 802.11 se propusieron una serie de procesos, que los datos deben llevar, los cuales se explican a continuación.

- **Scrambler:** Es el primer proceso que se realiza después de recibir el frame de la capa MAC, donde se cambia la posición de los bits mediante el polinomio generador (ecuación 3-7), evitando los '1' consecutivos y así tener un mayor ahorro de energía.

$$S(x) = x^7 + x^4 + 1 \tag{3-7}$$

El diagrama del scrambler se muestra en la figura 3.11.

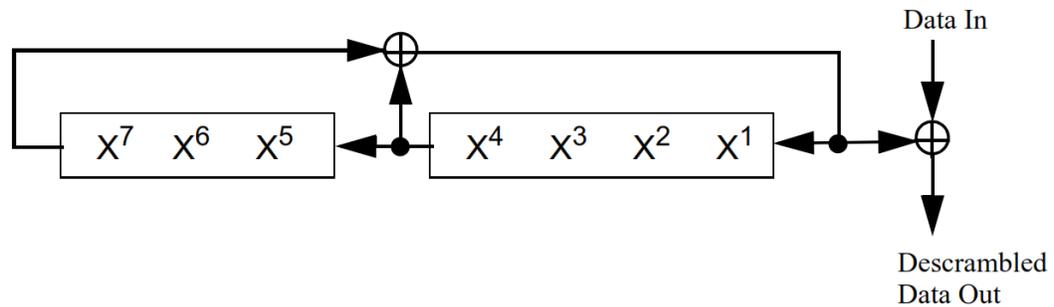


Figura 3.11 – Diagrama del Scrambler.

- **Codificador Convolutivo:** Es una técnica que se aplica en el transmisor para lograr detectar y corregir errores en la recepción de la información, donde se agregan bits de redundancia en base a dos polinomios generadores indicados en el estándar IEEE802.11g (133_8 y 171_8), esto genera una tasa de bits de 1/2, ya que se tienen 2 salidas (Figura 3.12), llamadas A y B. Para tener un mayor control del flujo de bits se agrega una etapa llamada puncturing la cual acorta las cadenas de bits obtenidas en la salida del codificador, obteniendo las tasas de 2/3 (Figura 3.13) y 3/4 (Figura 3.14).

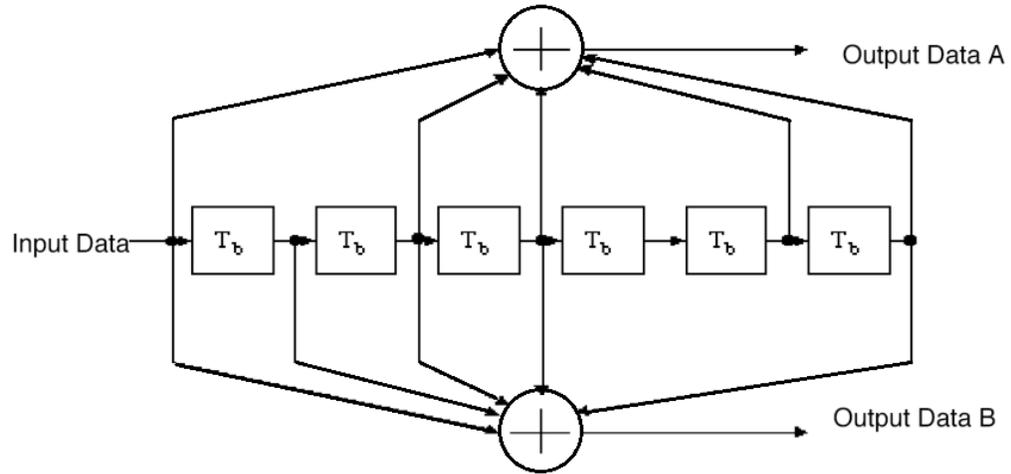


Figura 3.12 – Diagrama del Codificador Convolutivo.

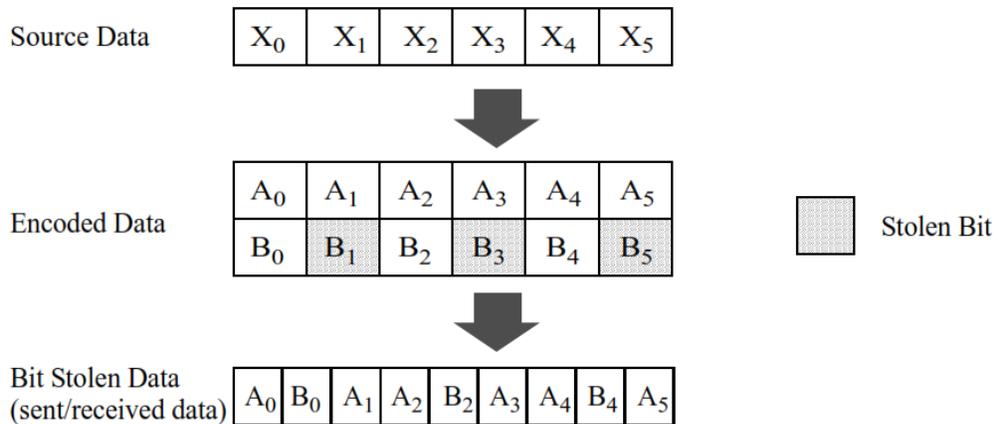


Figura 3.13 – Puncturing para una tasa de 2/3.

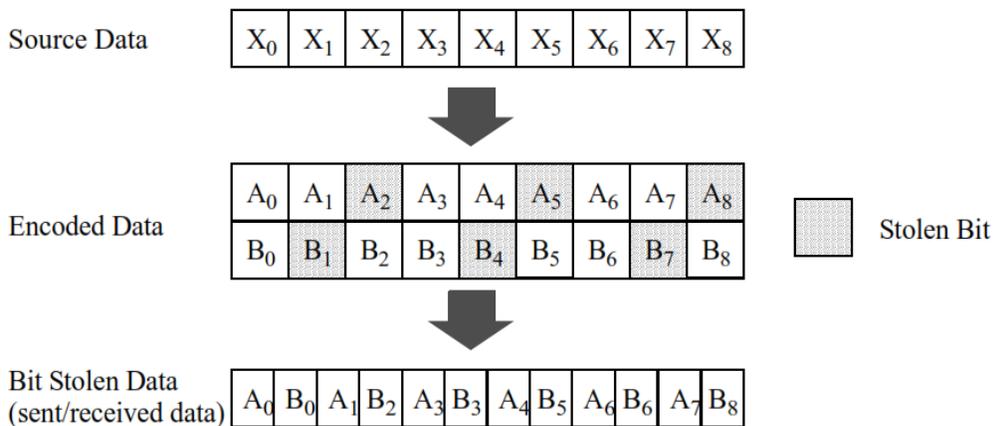


Figura 3.14 – Puncturing para una tasa de 3/4.

- **Interleaver:** Es la etapa de procesamiento donde los bits se intercalan, permitiendo que los errores no ocurran en bits consecutivos y así el Codificador Convolutivo logre corregir los errores que puedan generarse. De acuerdo al estándar IEEE 802.11 cada bit intercalado está en función de N_{CBPS} (Tabla 3.4) y su posición original.

Existen dos etapas de intercalamiento, la primera permite que los bits adyacentes sean transmitidos en subportadoras no adyacentes, donde i es la nueva posición del bit (Ecuación 3-8). La segunda etapa permite que los bits adyacentes codificados sean mapeados del menor al mayor bit significativo de la constelación, donde j es la nueva posición del bit (Ecuación 3-9).

$$i = (N_{CBPS}/16)(k \bmod 16) + \lfloor k/16 \rfloor \quad k = 0, 1, \dots, N_{CBPS} - 1 \quad (3-8)$$

$$j = s * \lfloor i/s \rfloor + (i + N_{CBPS} - \lfloor 16 * i/N_{CBPS} \rfloor) \bmod s \quad i = 0, 1, \dots, N_{CBPS} - 1 \quad (3-9)$$

- **Modulador:** Es la etapa donde los grupos de tamaño N_{CBPS} son mapeados en puntos de constelaciones BPSK, QPSK, 16-QAM o 64-QAM, por ejemplo las constelaciones que se pueden observar en la Figura 3.15.

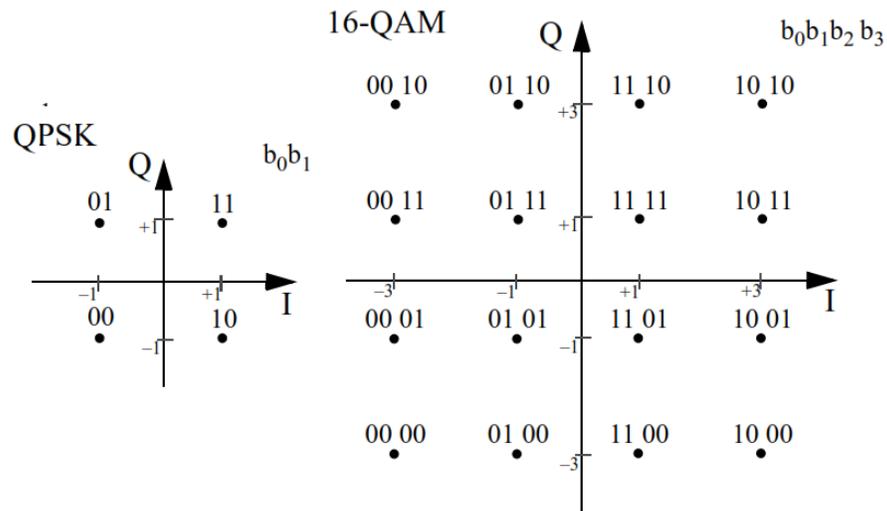


Figura 3.15 – Constelaciones de QPSK y 16-QAM.

Después de mapear los bits se normalizan los valores de acuerdo al factor K_{MOD} (Tabla 3.6), con el fin de obtener la misma potencia promedio en todos los mapeos.

Modulación	K_{MOD}
BPSK	1
QPSK	$1/\sqrt{2}$
16-QAM	$1/\sqrt{10}$
64-QAM	$1/\sqrt{42}$

Tabla 3.6 – Valores de normalización de las constelaciones.

- **Inserción de Portadoras Piloto:** Se agregan los valores 1, 1, 1, -1 en las posiciones -21, -7, 7, 21 del símbolo OFDM, cambiando su polaridad en cada transmisión, adicionalmente se agregan portadoras nulas en la posición de DC, es decir la portadora '0' y en los extremos de la posición -32 a la -27 y de la +27 a la +31.
- **IFFT:** Se aplica la IFFT en las 64 subportadoras para obtener el símbolo OFDM.
- **CP:** Se agrega el Ciclo Prefijo de 1/4 de la señal al final de cada símbolo.



En este capítulo se pudieron observar los aspectos y elementos de la capa MAC y capa PHY, necesarios para que exista una comunicación fiable en redes WLAN, de acuerdo al estándar IEEE 802.11g, utilizando el esquema OFDM, el cual permite alcanzar velocidades de hasta 54 Mb/s.

En la capa MAC y en la capa PHY, existen distintos elementos que ayudan a prevenir, detectar y corregir errores como el Interleaver, el CRC-32 y el Codificador Convolutacional, permitiendo que el Bit Error Rate sea el menor posible.



Capítulo 4

USRP N210

Introducción

La importancia del SDR en los últimos años ha aumentado, ya que la tecnología ha logrado crear equipos los cuales realmente sean fiables, ya que su implementación puede cubrir las exigencias de la sociedad actual, es decir tienen capacidades comerciales.

Ettus Research fue una de las primeras compañías en vender SDR para cualquier persona que lo solicite, con la cualidad de ofrecer equipos modulares que se adapten a la necesidad del usuario que lo adquiera, en caso particular, el USRP N210 tiene opción de utilizar una tarjeta llamada SBX que permite al dispositivo trabajar en frecuencias de Wi-Fi, razón por la cual se eligió utilizar éste equipo.

Este capítulo se centrará en describir que es el equipo USRP N210, que partes lo conforman, la importancia del firmware y su utilidad dentro del proyecto de tesis.

4.1 Características Generales

El USRP N210 está conformado por distintos componentes los cuales permiten cambiar la frecuencia en base a las órdenes que se le indiquen, así como pasar los datos de banda base a una frecuencia, para que así éstos puedan ser transmitidos por el aire.

En la Figura 4.1 se puede observar el diagrama a bloques, el cual describe el funcionamiento del USRP N210, tanto para transmisión, como para recepción.

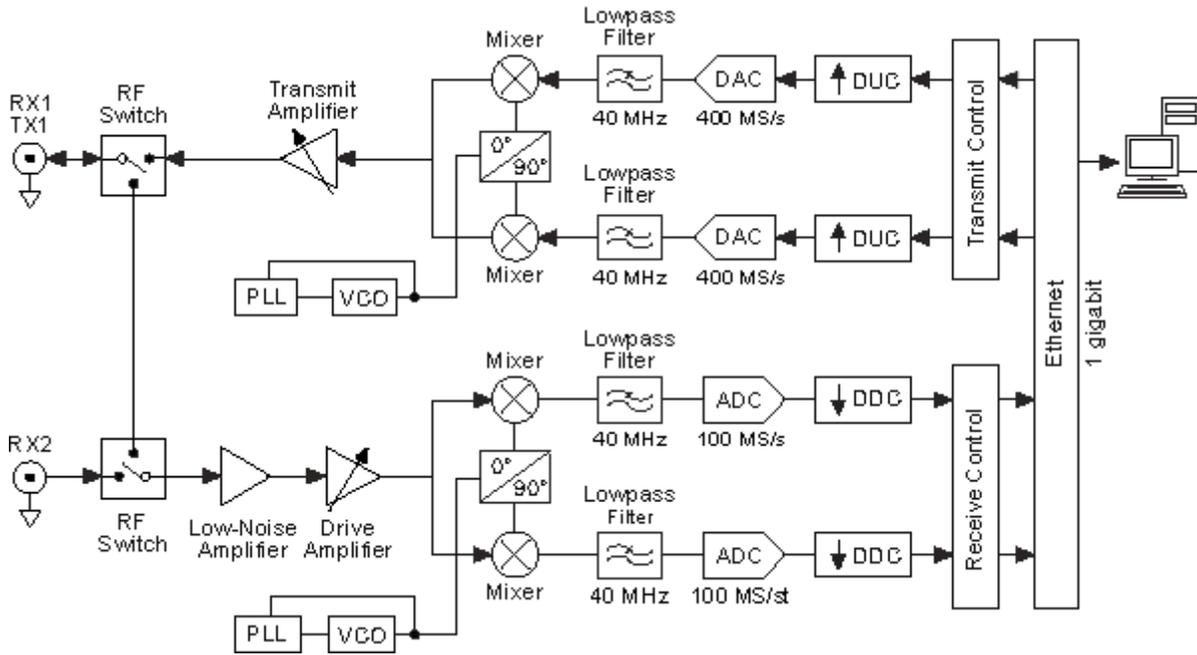


Figura 4.1 – Diagrama a bloques de los componentes del USRP.

Los distintos componentes del USRP N210 se describen a continuación.

Ethernet: Es el puerto de conexión entre un ordenador y el USRP. Para el USRP N210 la velocidad máxima que puede alcanzar ésta interfaz es de 25 MS/s a 16 bit de I/Q.

Transmit Control: Es un módulo el cual indica las características de transmisión que el equipo debe tener.

Digital Up Converter (DUC): Es un conversor ubicado en el transmisor, el cual aumenta la tasa de transferencia de datos, para que puedan ser transmitidos análogamente de acuerdo a la velocidad de muestreo que indique el usuario.

Digital Analog Converter (DAC): Es un conversor ubicado en el transmisor, el cual cambia los valores digitales (1 y 0) a valores analógicos de voltaje, para que la información pueda ser transmitida en el aire imitando la señal analógica que el usuario implemente, montando la señal en una frecuencia intermedia.



Low Pass Filter (LPF): Es filtro que se aplica para permitir solamente el paso de la señal en la frecuencia intermedia que se montó previamente.

Mezclador: Es donde la señal de frecuencia intermedia se pasa a la frecuencia programada en el Oscilador Controlado por Voltaje (VCO) y a la fase que el Lazo de Seguimiento de Fase(PLL) indique, también se inserta un segundo canal en caso de ser utilizado.

Transmit Amplifier: Es la etapa donde la señal recibe ganancia para que pueda viajar en el aire, la potencia máxima del amplificador, depende del modelo y la tarjeta hija que se utilice.

RF Switch: Conmuta entre las entradas y salidas del USRP, para que puedan ser procesados si esta en modo de transmisión o de recepción.

Low Noise Amplifier: Es una etapa de preamplificación que se encuentra en el receptor del USRP, la cual ayuda a que la señal recibida pueda ser amplificada correctamente en una etapa posterior.

Drive Amplifier: Amplifica la señal preamplificada para que pueda ser tratada correctamente en las etapas posteriores.

Analog Digital Converter (ADC): Es un conversor ubicado en el receptor, el cual interpreta las señales analógicas recibidas en bits.

Digital Down Converter (DDC): Es un conversor ubicado en el receptor, el cual reduce la tasa de transferencia de datos, para que puedan ser procesados en banda base.

Receive Control: Es un módulo el cual indica las características de recepción que el equipo debe tener.

4.2 Motherboard y Daughterboard

Físicamente el procesamiento en banda base es llevado a cabo en la llamada Motherboard, la cual es una FPGA, la cual es programada de acuerdo a los parámetros dados por el usuario, para así preparar la señal digitalmente, los modelos de FPGA varían dependiendo del USRP que se utilice, como se puede observar en la Tabla 4.1.

Modelo de USRP	Ancho de Banda de Procesamiento para el ADC (MS/s)	Ancho de Banda de Procesamiento DAC (MS/s)
USRP B100	64 MS/s	128 MS/s
USRP 1	64 MS/s	128 MS/s
USRP E100/E110	64 MS/s	128 MS/s
USRP B200/B210	61.44 (simplex)	61.44 (simplex)
USRP N200/N210	100 MS/s	100 MS/s
USRP X300/X310	200 MS/s	200 Ms/s

Tabla 4.1 – Velocidades de procesamiento de las Motherboards de los USRP.

Como se puede observar en la Tabla 4.1, específicamente para el USRP N210 la velocidad máxima de procesamiento del ADC y del DAC es de 100 MS/s, aunque cabe destacar que es velocidad por los dos puertos que tiene el equipo, es decir cada puerto tiene una velocidad máxima de 50 MS/s.

En la motherboard se conecta la llamada daughterboard, la cual es una tarjeta, que lleva los valores digitales que procesa la motherboard a valores analógicos. Existen distintos tipos de daughterboard que pueden utilizarse, como se puede observar en la Tabla 4.2, con el fin de que el equipo pueda trabajar en los diferentes rangos de frecuencias de las distintas tecnologías inalámbricas existentes.

Daughterboard	Rango de Frecuencias	Ancho de Banda Analógico
WBX-120	50 MHz – 2.2 GHz	120 MHz
SBX-120	400 MHz – 4.4 GHz	120 MHz
CBX-120	1.2 GHz – 6 GHz	120 MHz
WBX	50 MHz – 2.2 GHz	40 MHz
SBX	400 MHz – 4.4 GHz	40 MHz
CBX	1.2 GHz – 6 GHz	40 MHz
TVRX2	50 MHz – 860 MHz	Configurable – 1.7 to 10 MHz
DBSRX2	800 MHz – 2.3 GHz	Configurable – 8 to 80 MHz
BasicRX/BasicTX	1 – 250 MHz	*Determinado pro la frecuencia de maestro del ADC/DAC. Es requerido un filtro externo.
LFRX/LFTX	DC-30 MHz	30 MHz

Tabla 4.2 – Modelos de Daughterboards para los USRP.

Dependiendo del modelo de USRP que se utilice, es la cantidad de daughterboards que se pueden instalar, por ejemplo en los modelos X300/X310 pueden instalarse hasta dos daughterboards, pero en los modelos N200/N210, puede instalarse solo una daughterboard.

4.3 Conexión con el USRP N210

Para trabajar con el USRP N210 se deben de seguir una serie de pasos, para que exista conexión entre un ordenador y el equipo USRP, el cual puede trabajar con dos sistemas operativos, Windows y Linux, variando el rendimiento y la forma de programación del USRP, dependiendo del sistema operativo seleccionado.

En caso de seleccionar Linux para trabajar con el USRP, se debe tener en cuenta que la instalación es complicada, ya que hay que preparar el sistema operativo con distintos requisitos para que pueda instalar el “UHD”, el cual es el controlador que comunica los puertos del USRP, con los puertos del ordenador para que exista un conexión entre ellos. Una vez instalada la versión de UHD requerida, se puede trabajar con el USRP utilizando GNU Radio (Figura 4.2) o algún otro software para Linux que trabaje con UHD.

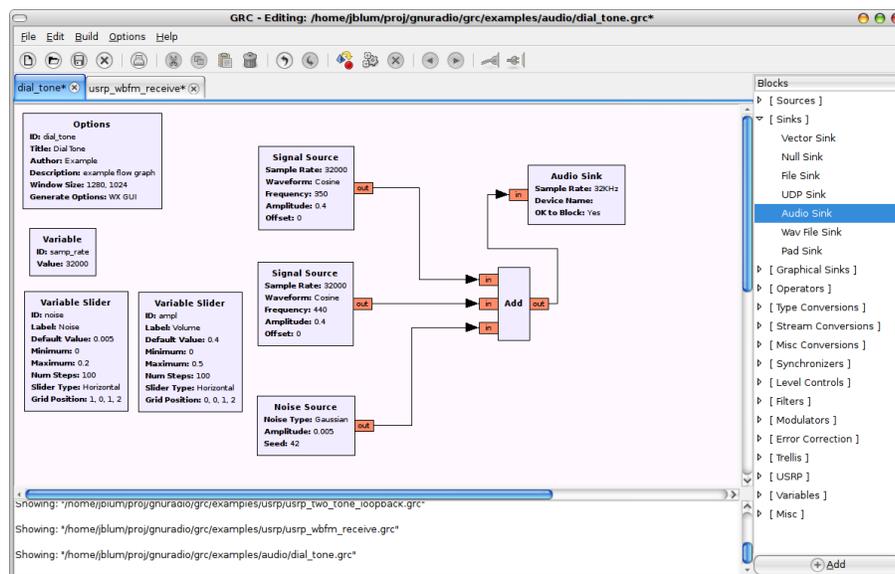


Figura 4.2 – GNU Radio Companion.

La instalación de los controladores para Windows es más sencilla, ya que solo se descargan los softwares necesarios proporcionados por National Instruments o Matlab

y el Equipo USRP se podrá comunicar con el ordenador que trabaje con Windows, utilizando los respectivos bloques dentro de LabVIEW o Matlab.

Las limitantes de usar los USRP en Windows, es que solo existen dos programas que trabajen con éstos equipos de SDR, como se mencionó anteriormente son LabVIEW (Figura 4.3) y Matlab (Figura 4.4), además existen parámetros que no se pueden cambiar como el tamaño del búfer de datos que existe en la Interfaz Ethernet, siendo configurable en Linux.

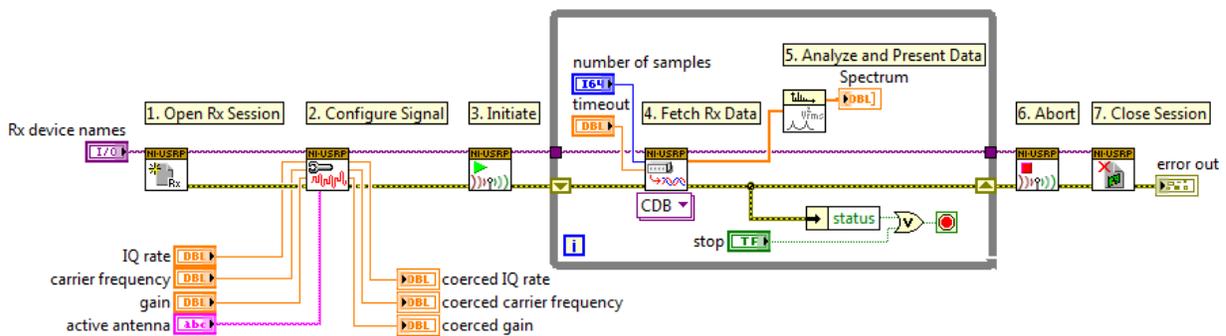


Figura 4.3 – LabVIEW con bloques de USRP.

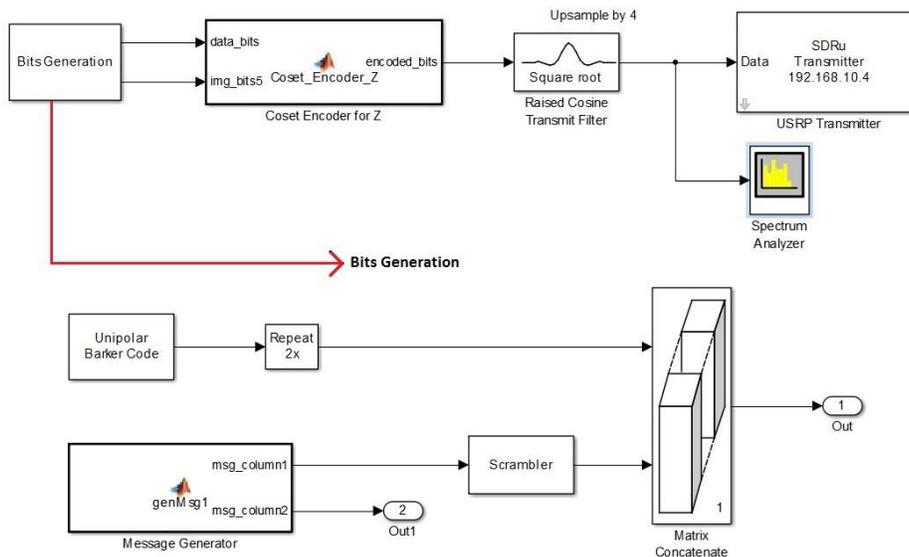


Figura 4.4 – Matlab con bloques de USRP.



En Windows se puede trabajar de una manera más amigable con los equipos USRP, ya que además de su sencilla instalación, los bloques de control de LabVIEW y Matlab, son bastante estables y fiables, lo que ayuda a los USRP a mantener su rendimiento en la mayor capacidad posible.

Es muy importante después de lograr comunicación con los USRP, ya sea en Linux o Windows, mantener la versión de Firmware y de FPGA lo más actualizada posible, ya que se corrigen errores en cada versión que impactan directamente al rendimiento del equipo. Dependiendo del modelo de USRP que se utilice, es la forma de actualizar el Firmware y el FPGA, que puede ir desde utilizar un cable USB a copiar los archivos a una memoria SD, aunque para el modelo N210 se tiene que hacer a través de la interfaz Ethernet ya sea utilizando el programa *usrp_n2xx_net_burner_gui.py* en Linux o “*NI-USRP Configuration Utility*” en Windows.

Conocer a la perfección el funcionamiento del equipo SDR que se desee utilizar es de gran importancia si se desea trabajar con él en algún proyecto, ya que dependiendo del equipo elegido será el alcance del proyecto a realizar.

El USRP N210 tiene la velocidad necesaria de procesamiento la cual es 50 MS/s, aunque tiene una limitante en la interfaz Ethernet de conexión, lo que puede provocar que no se puede transferir la totalidad de datos procesados o a procesar por del FPGA al ordenador.



Capítulo 5

Descripción del Código

Introducción

El estándar IEEE 802.11 al tener una gran serie de pasos para lograr enviar un frame a través del aire, provoca que en caso de querer programarlo se cree un código relativamente grande, pero implementarlo en un lenguaje como LabVIEW ayuda a visualizar el código de una manera reducida, creando programas más cortos.

LabVIEW al ser un lenguaje de programación amigable, ayuda a crear códigos de programación que resuelvan problemas y representen algoritmos con pasos reducidos, ayudando a que el programa total resultante sea corto a comparación de otros lenguajes de programación y entendible, ya que las operaciones realizadas se pueden deducir al seguir el diagrama a bloques construido.

5.1 Frame de Capa MAC

Para poder verificar que los módulos programados correctamente, se tomó la decisión de seguir exactamente cada paso del ejemplo que viene en el estándar IEEE 802.11 [1], donde el primer paso es tener un frame con características de capa MAC para que el frame resultante al pasar por todo el procesamiento de capa PHY sea lo más real posible.



5.1.1 Valores Iniciales

En la Figura 3.9 se puede observar la estructura de un frame de capa MAC y de acuerdo con el ejemplo del estándar IEEE 802.11 los campos contienen los siguientes valores en hexadecimal (Solo el campo de datos o Frame Body se mostrará en caracteres para una mejor apreciación).

Frame control: 0204

Duration/ID: 2E00.

Address 1: 0060 08CD 37A6.

Address 2: 0020 D601 3CF1.

Address 3: 0060 08AD 3BAF.

Sequence Control: Se aumenta de uno en uno por cada frame enviado hasta 4096.

Frame Body:

Joy, bright spark of divinity,

Daughter of Elysium,

Fire-insired we trea

En código de LabVIEW se pueden observar estos valores, como se muestra en las Figuras 5.1 y 5.2.

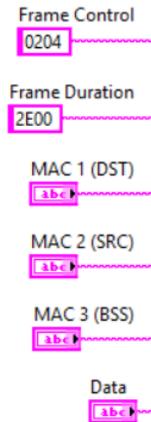


Figura 5.1 – Datos iniciales en el diagrama a bloques de LabVIEW.

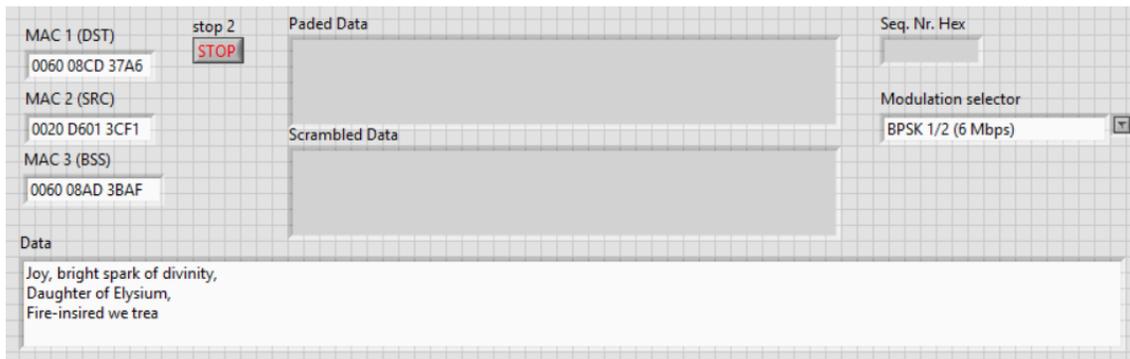


Figura 5.2 – Datos iniciales en el panel frontal de LabVIEW.

5.1.2 CRC-32

Una vez que se tienen todos los campos del frame se procede a llenar el campo de Frame Check Sequence, con un Código de Redundancia Cíclica (CRC), el cual utiliza un polinomio generador de 32 coeficientes, como se muestra en la Fórmula 3-1.

El CRC-32 genera un valor de 4 bytes de una cadena de n elementos, mediante una división polinomial, donde los datos son divididos entre un polinomio generador. Para lograr esto en LabVIEW, el programa se incluyó en un bloque (Figura 5.3), el cual se dividió en dos partes, primero se genera un registro de corrimiento en base al polinomio generador que se indique (Figura 5.4) y después se va dividiendo la información

mediante un compuerta OR Exclusiva (Figura 5.5), guardando el residuo en un arreglo vacío hasta llegar al resultado final de 4 bytes.



Figura 5.3 – Bloque de CRC-32 en LabVIEW.

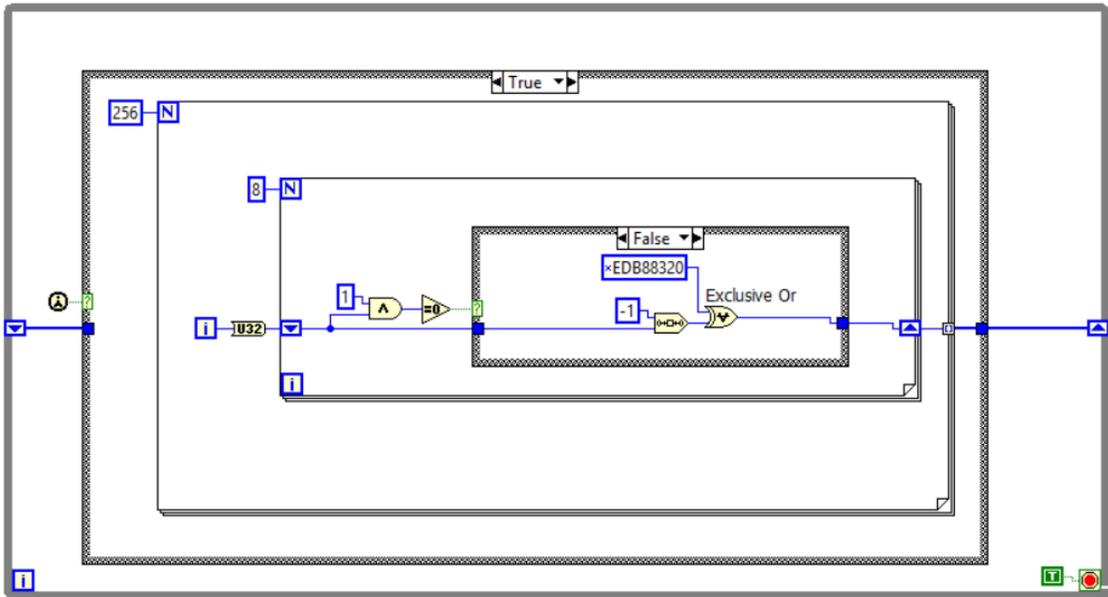


Figura 5.4 – Primera etapa del CRC-32.

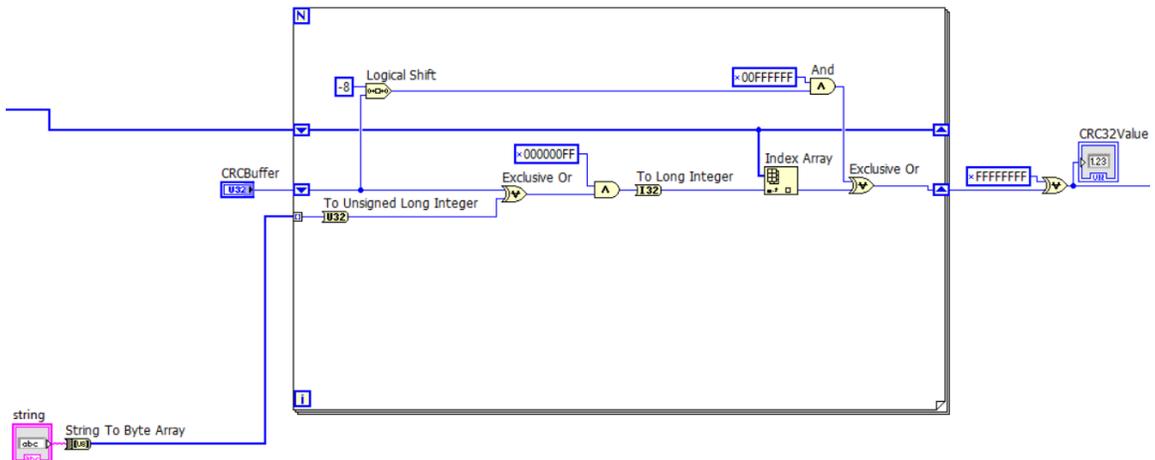


Figura 5.5 – Segunda etapa del CRC-32.



El resultado del CRC-32 se obtiene en una variable numérica, para que este valor sea igual al del frame de capa MAC generado anteriormente, se convierte a carácter y se agrega al final del frame.

5.2 Frame de Capa PHY

Una vez que se tiene en frame de capa MAC, se tienen que agregar los encabezados de capa física (Figura 3.10), ya que en éstos encabezados se encontraran los campos de la señal, que corresponden a la modulación a utilizar, el tamaño del frame, el bit de paridad y bits reservados para posterior uso, también se agregan campos a los datos para su posterior procesamiento.

La generación del frame de capa PHY se programó en un bloque llamado *Generate_Bits*, el cual se puede dividir en tres partes.

5.2.1 Selector de Modulación

Muchos de los procesos de transmisión, como el padding, el Interleaver e inclusive la generación del símbolo OFDM, dependen de la modulación que se vaya a ocupar para la sección de datos del frame de capa PHY. Por ésta razón se decidió incluir todos los parámetros en un case de 8 opciones (Figura 5.6), el cual contenga y entregue todos los valores dependientes del tipo de modulación.

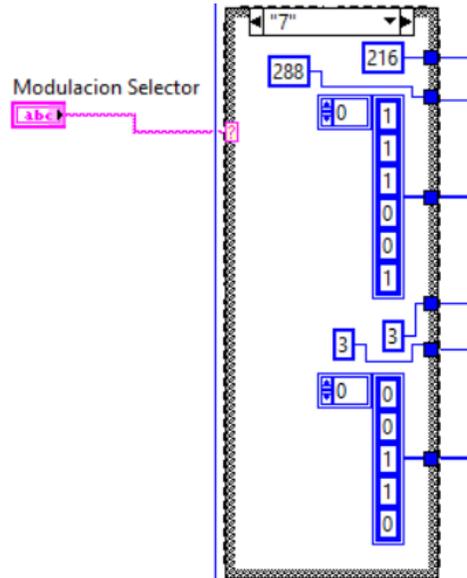


Figura 5.6 – Selector de Modulaci3n.

En el selector de modulaci3n se incluyen los campos que se explican a continuaci3n.

- **N_{DBPS}**: Valor mostrado en la Tabla 3.4, utilizado para generar los bits de padding, de las F3rmulas 3-4 y 3-5.
- **N_{CBPS}**: Valor mostrado en la Tabla 3.4, utilizado en las permutaciones del Interleaver, como se muestra en las F3rmulas 3-8 y 3-9.
- **Puncturing Patern**: Cadena de bits utilizada por la etapa de Puncturing, la cual indica cuantos bits retirar de la cadena de datos.
- **Indicador de Modulaci3n**: Valor que va del 0 al 7 el cual le indica al modulador que modulaci3n utilizar.
- **Rate**: Valor mostrado en la Tabla 3.3, el cual se inserta en el campo de Rate del encabezado, para que el receptor conozca la modulaci3n utilizada en los datos.

5.2.2 Padding

Una vez que se conoce la modulación a utilizar y el valor de N_{DBPS} , se procede a utilizar las Fórmulas 3-4, 3-5 y 3-6, como se muestra en la Figura 5.7, donde se obtiene la cantidad de '0' que se agregarán a los datos, además se agregan los bits de servicio y cola.

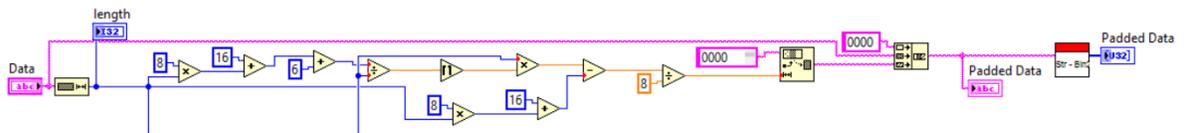


Figura 5.7 - Etapa de Padding.

Finalmente se obtiene la longitud de la cadena después del padding y la inserción de los bits de servicio y cola, valor que se agregará al campo de longitud del encabezado de capa PHY, es decir la sección correspondiente a la señal del frame de capa PHY.

5.2.3 Bit de Paridad

Por último se obtiene el bit de paridad (Figura 5.8), sumando la cantidad de '1' existentes en la sección de Señal del frame, agregando 1 o 0, en el campo de Paridad del frame, si es par o impar el valor resultante.

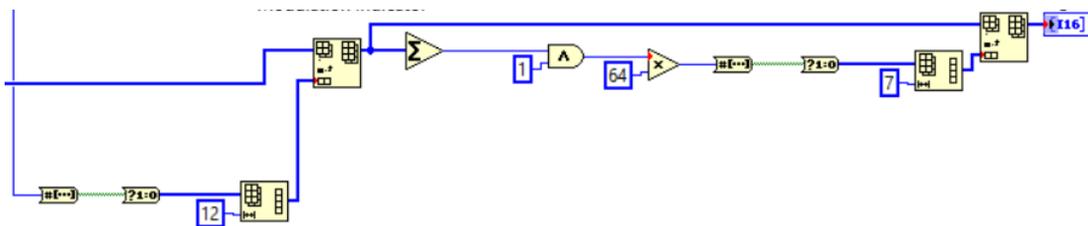


Figura 5.8 – Generación del bit de paridad.

Después de obtener el bit de paridad, se unen todos los valores correspondientes al apartado de Señal del frame.

5.3 Scrambler

La siguiente etapa es la de Scrambling, en donde se cambia la posición de los bits, para evitar ráfagas de '1' consecutivos y así ahorrar energía en la transmisión. El código de LabVIEW se realizó en base a la Figura 3.11.

El código se incluyó en un bloque llamado *Scrambler* (Figura 5.9), el cual contiene registros de corrimiento, los cuales están ordenados de acuerdo al polinomio generador de la Fórmula 3-7 (Figura 5.10), realizando las sumas módulo dos con compuertas XOR, dentro de un for que termina hasta que su contador sea igual al tamaño de la cadena de entrada.

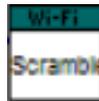


Figura 5.9 – Bloque del Scrambler en LabVIEW.

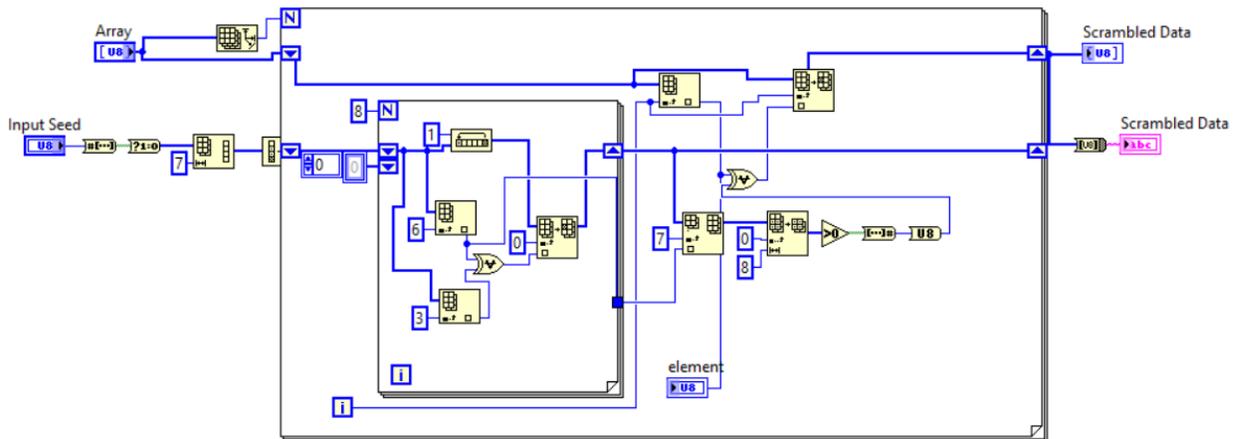


Figura 5.10 – Diagrama a bloques del Scrambler en LabVIEW.

Aunque los datos de entrada son numéricos, por la forma de trabajar de LabVIEW fue más sencillo obtener el resultado del scrambler en hexadecimal en una cadena de caracteres.

La etapa de scrambler solo se realiza para los datos de acuerdo al estándar IEEE 802.11, por lo que la sección de señal del frame no pasa por esta etapa.

5.4 Codificación Convolutiva

La codificación convolutiva se aplica para que el sistema tenga la capacidad de detectar y corregir errores, en este caso para corregir un error de cada ocho bits.

En LabVIEW existe un bloque llamado *MT Convolutional Encode.vi* (Figura 5.11), el cual realiza la codificación convolutiva de una manera sencilla, ya que solo se tienen que dar los polinomios a utilizar en octal y la venta k para que realice la codificación. Los polinomios utilizados fueron 133_8 y 171_8 con una $k=8$.

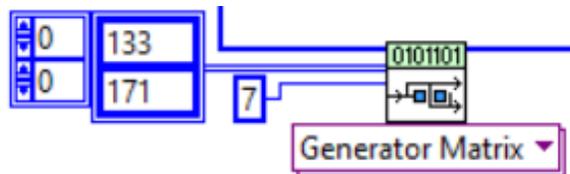


Figura 5.11 – Codificador Convolutiva de LabVIEW.

Esta etapa se aplica tanto a la señal como a los datos, donde la señal se tomó directamente del bloque *Generate_Bits*, ya que los datos se encontraban de manera numérica, que es el tipo de entrada que pide el bloque de codificación. Los datos, por otra parte, salen del scrambler como carácter y al no existir un conversor carácter-numérico en LabVIEW, se programó una etapa previa en un bloque llamado *String_to_Bin* (Figura 5.12), transformando los datos tipo carácter a numérico, como se muestra en la Figura 5.13.



Figura 5.12 – Bloque String_to_Bin de LabVIEW.

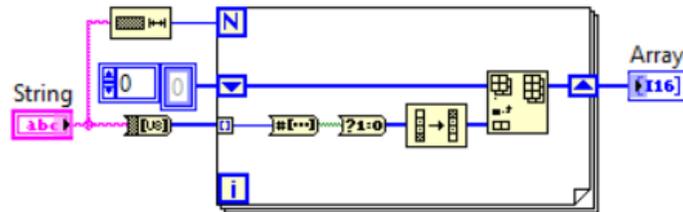


Figura 5.13 – Diagrama a Bloques del bloque String_to_Bin.

Para transformar de carácter a numérico se tomó cada carácter de la cadena entrante y se aplicó un for el cual toma esos caracteres, los transforma a binario y una vez conociendo su valor binario se puede reinterpretar como un número.

5.5 Puncturing

El acortamiento de los bits es una etapa exclusiva de la sección de datos del frame de capa PHY, ya que es la que se encarga de cambiar la tasa de bits inicial proveniente del codificador convolucional, la cual es de $1/2$, a nuevas tasas que pueden ser $2/3$ o $3/4$.

Para realizar el puncturing en LabVIEW se programó el bloque *Puncturing* (Figura 5.13), el cual recibe el patrón de acortamiento desde el bloque *Generate_Bits* en un for, donde entran los bits de información provenientes del codificador convolucional y son puestos en un nuevo arreglo solo los bits que el patrón indique, el programa de Puncturing se puede observar en la Figura 5.14.



Figura 5.14 – Bloque Puncturing de LabVIEW.

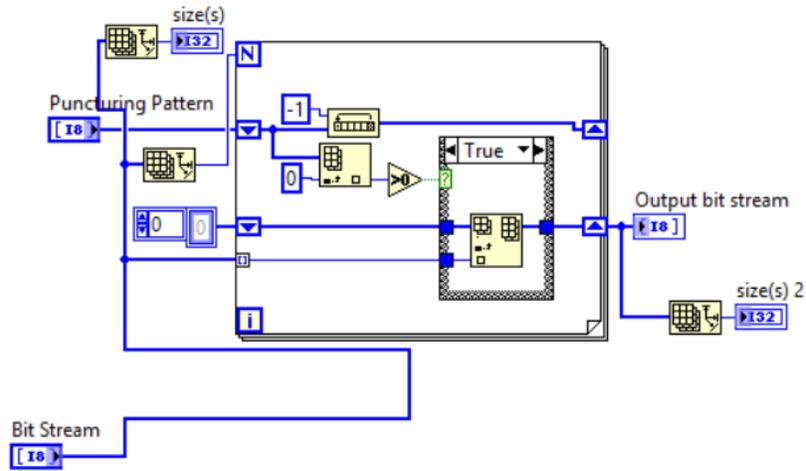


Figura 5.15 – Diagrama a bloques del bloque Puncturing.

EL tipo de valor de salida del bloque es numérico.

5.6 Interleaver

La etapa de Interleaver es la encargada de evitar adyacencias entre bits y portadoras, con el fin de que los errores ocurran aisladamente. El intercalamiento de bits se aplica tanto para los datos como para la señal.

En LabVIEW esto se logra a través de las fórmulas 3-8 y 3-9, las cuales se programaron en el bloque llamado *Interleaver* (Figura 5.16), donde las dos permutaciones son programadas en dos ciclos for. El tamaño de los ciclos dependen de N_{BPSC} y s , valores enviados del bloque *Generate_Bits*. El programa de Interleaving se puede observar en la Figura 5.17.

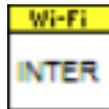


Figura 5.16 – Bloque Interleaver de LabVIEW.

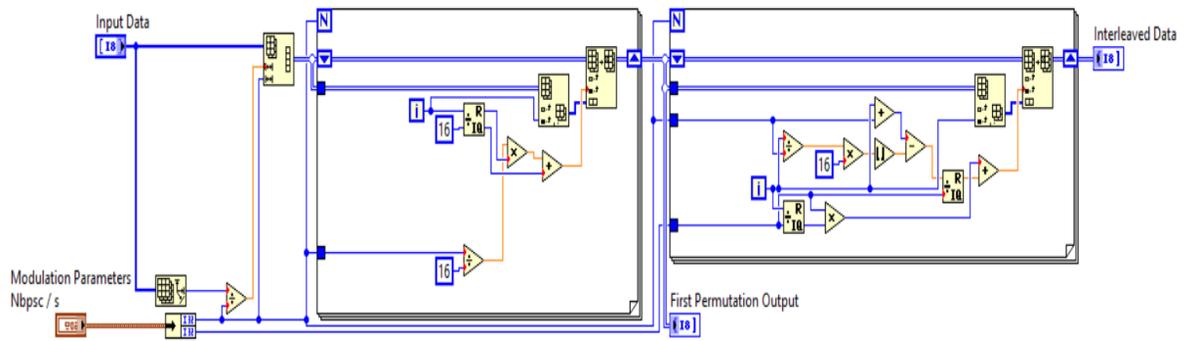


Figura 5.17 – Diagrama a bloques del bloque Interleaver.

Finalmente la información se agrupa en una matriz de 48 elementos por la cantidad de símbolos a enviar, es decir se hace una conversión serial-paralelo.

5.7 Modulador

En la modulación se hace el cambio de información digital a analógica, para que el frame pueda ser amplificado y enviado a través del aire, donde las modulaciones utilizadas son BPSK, QPSK, 16-QAM y 64-QAM.

La modulación de los datos se realizó en el bloque llamado *ModData* (Figura 5.18), el cual contiene un case de ocho opciones, donde el indicador de modulación permite escoger entre las cuatro modulaciones utilizadas en Wi-Fi. Dentro del case se encuentra el bloque llamado *MT Map Bit to Symbols VI* (Figura 5.19), el cual en base a una constelación relaciona un bit o una serie de bits con un punto de esa constelación, regresando un arreglo con un valores complejos.

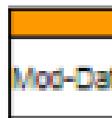


Figura 5.18 – Bloque ModData de LabVIEW.



Figura 5.19 – Bloque MT Bit to Symbols VI de LabVIEW.

La constelación que ingresa al bloque *MT Map Bit to Symbols VI*, se tomaron de las propuestas por el estándar IEEE 802.11 [1] y se normalizaron de acuerdo a la Tabla 3.6.

La sección de señal del frame se moduló en un bloque diferente llamado *ModSignal* (Figura 5.20), ya que siempre utiliza la modulación BPSK 1/2. El programa completo del Modulador se puede observar en la Figura 5.21 para los datos y en la Figura 5.22 para la señal.

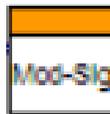


Figura 5.20 – Bloque ModSignal de LabVIEW.

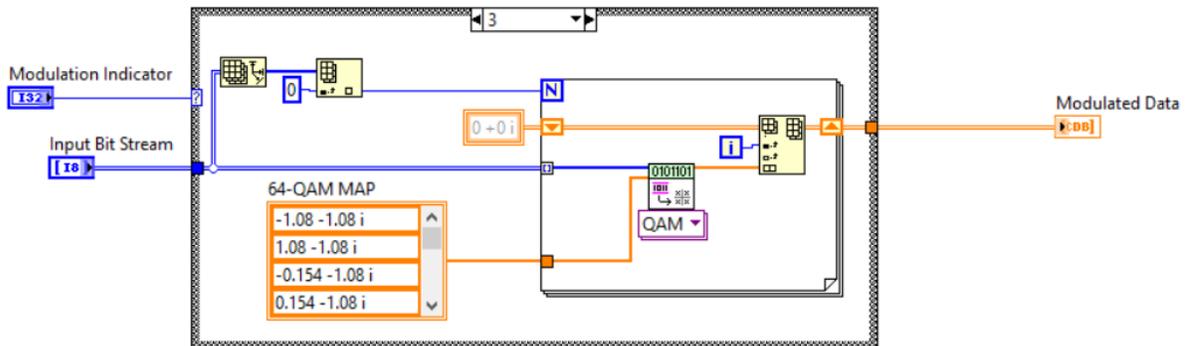


Figura 5.21 – Diagrama a bloques del bloque ModData.

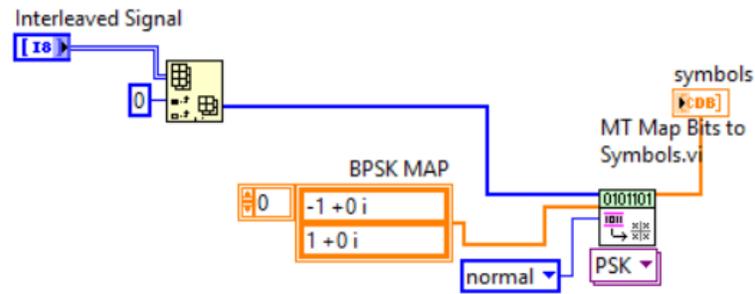


Figura 5.22 – Diagrama a bloques del bloque ModSignal.

Los datos de entrada del bloque son tipo matriz numérica y los datos de salida son datos tipo matriz compleja.

5.8 Pilotos

Como se mencionó en el subtema 3.3.2, después de modular se realiza una inserción de pilotos en las posiciones -21, -7, 7, 21 del símbolo OFDM con valores de 1, 1, 1, -1 respectivamente, además la polaridad de los pilotos cambia su por cada símbolo enviado. Para realizar ésto en LabVIEW se programaron los bloques *InsertPilots* (Figura 5.23) e *InsertPilotSignal* (Figura 5.24), en donde *InsertPilotSignal* utiliza siempre el primer patrón de polaridad e *InsertPilots* varía la polaridad a partir del segundo patrón, que corresponde a la sección de datos del frame de capa PHY.



Figura 5.23 – Bloque InsertPilots de LabVIEW.



Figura 5.24 – Bloque InsertPilotSignal de LabVIEW.

Como se observa en la Figura 5.25, el bloque InsertPilots, mediante un ciclo for, toma los valores correspondientes a los pilotos, después de multiplicarlos por el patrón de polaridad y los inserta en las posiciones correspondientes. Lo mismo ocurre para la sección de señal (Figura 5.26), solo que sin el ciclo for ya que siempre se utiliza el mismo patrón en el bloque InsertPilotSignal.

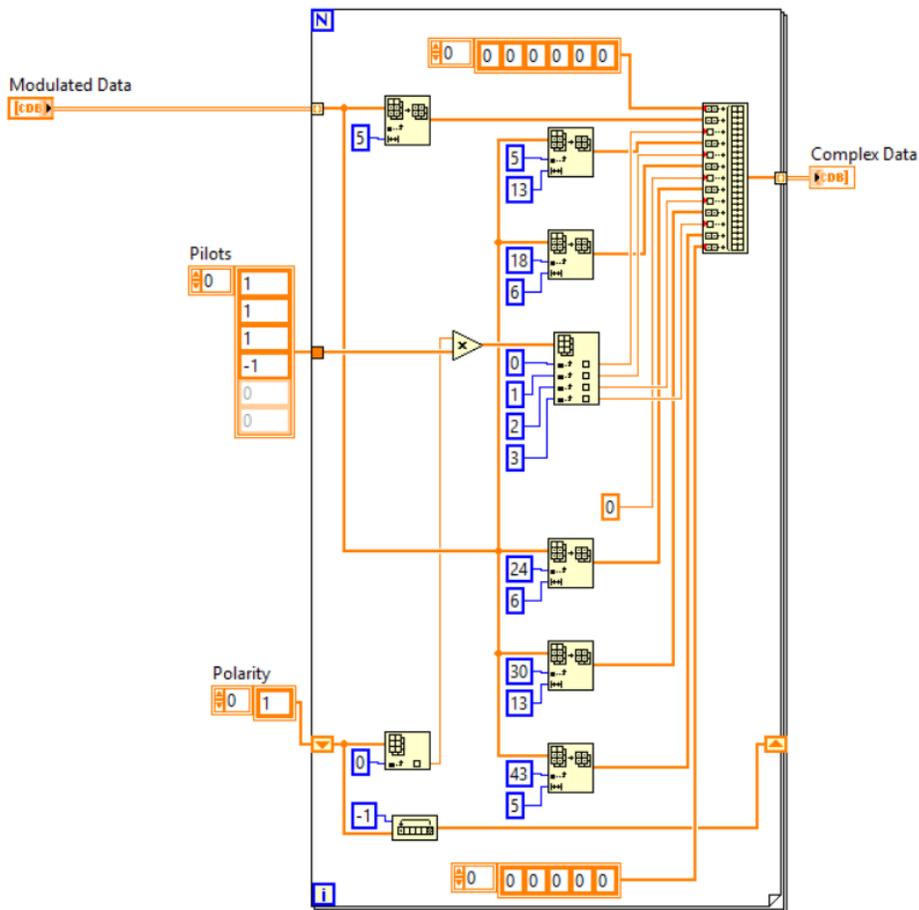


Figura 5.25 – Diagrama a bloques del bloque InsertPilots.

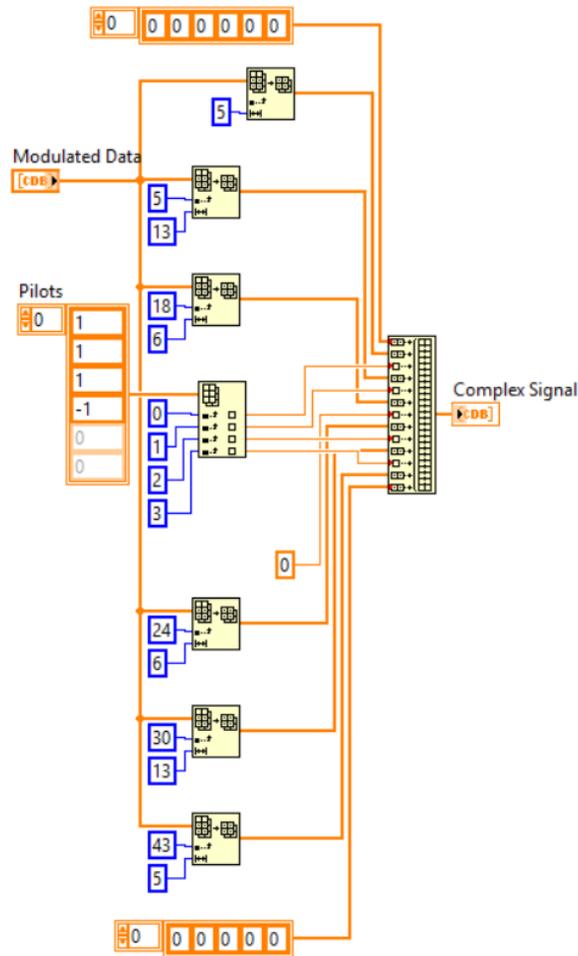


Figura 5.26 – Diagrama a bloques del bloque InsertPilotSignal.

En los bloques utilizados para la inserción de pilotos, los datos de entrada son de tipo matriz compleja y los datos de salida también son de tipo matriz compleja.

5.9 IFFT y Ciclo Prefijo

Una de las últimas secciones del código es la IFFT, en esta etapa cambia del plano en frecuencia al plano en tiempo, para generar así los símbolos OFDM que serán transmitidos a través del aire.

Para realizar la IFFT en LabVIEW se utilizó el bloque *Inverese FFT VI* (Figura 5.27), el cual solo pide los datos a transformar y hay que indicar si el símbolo llevará componente de DC. Ya que la IFFT se realiza por símbolo en los datos fue necesario un for el cual permita al bloque de IFFT realizar la operación correctamente, en la señal no fue necesario el for ya que solo es un símbolo, en la Figura 5.28 se puede apreciar cómo se realiza la IFFT para los Datos y la Señal.



Figura 5.27 – Bloque Inverse FFT VI de LabVIEW.

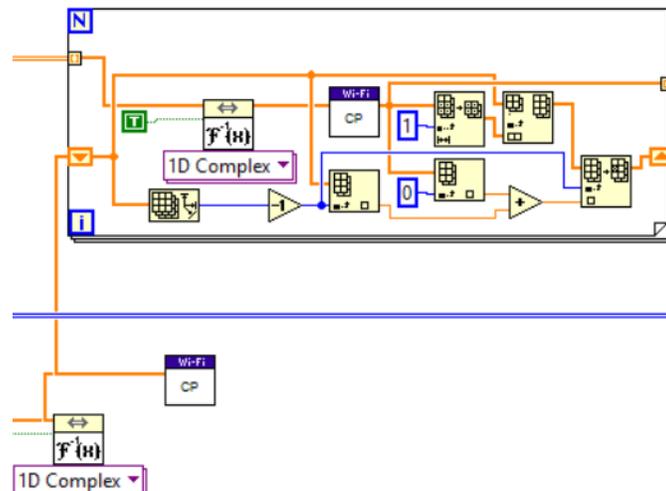


Figura 5.28 – Diagrama a bloques de la IFFT para los Datos y para la Señal.

Además en la figura 5.26 se puede observar el bloque *CP* (Figura 5.29), el cual toma un cuarto de la señal y lo inserta al principio del frame en tiempo, el cual se utiliza para compensar las multitrayectoras que puedan existir en la transmisión.



Figura 5.29 – Bloque CP de LabVIEW.

5.10 Secuencias de Entrenamiento

El siguiente paso es el de insertar la Long Training Sequence y la Short Training Sequence descritas en las Fórmulas 3-2 y 3-3, las cuales serán utilizadas en el receptor para la sincronización, el alineamiento de símbolo y la ecualización.

La inserción de las secuencias se realiza en bloque *InsertPreamble* (Figura 5.30), en donde se insertan las secuencias correspondientes y se traslapan de acuerdo a estándar IEEE 802.11, además de ser repetidas 2.5 veces, el diagrama a bloques que realiza la inserción de los preámbulos se puede observar en las Figuras 5.31 y 5.32.



Figura 5.30 – Bloque InsertPreamble de LabVIEW.

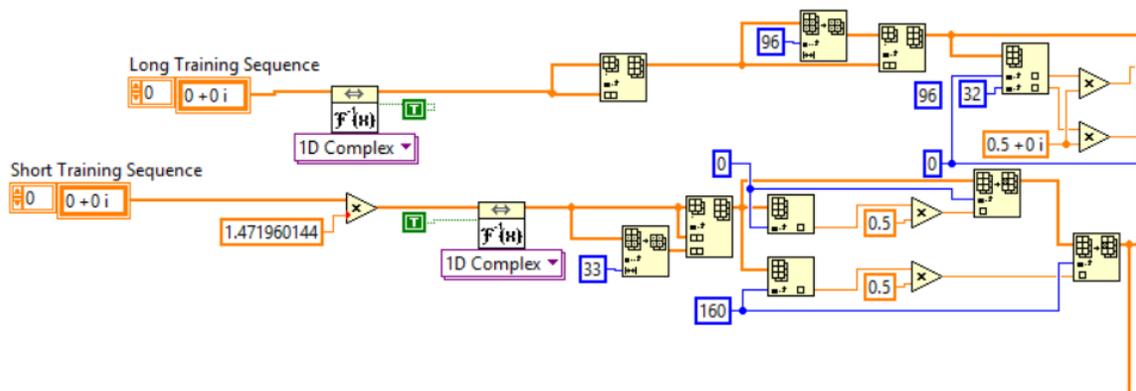


Figura 5.31 – Diagrama a bloques del bloque InsertPreamble.

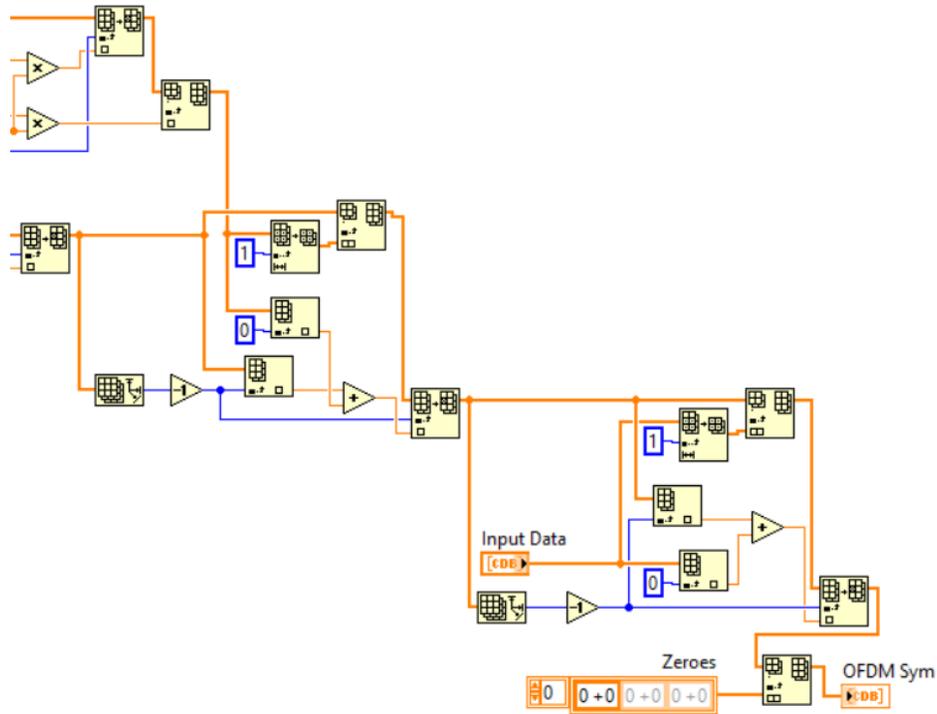


Figura 5.32 – Diagrama a bloques del bloque InsertPreamble (Continuación).

5.11 Conexión al USRP y Herramientas para un Posterior Análisis

En esta etapa, al tener el frame listo para la transmisión, se agregan los bloques necesarios para que LabVIEW pueda comunicarse con el USRP N210, estos bloques se describieron en el subtema 2.2.3, además se agrega una sección donde se genera un archivo para realizar análisis de BER, los bloques de conexión de USRP y generación de archivos se pueden observar en las Figuras 5.33 y 5.34.

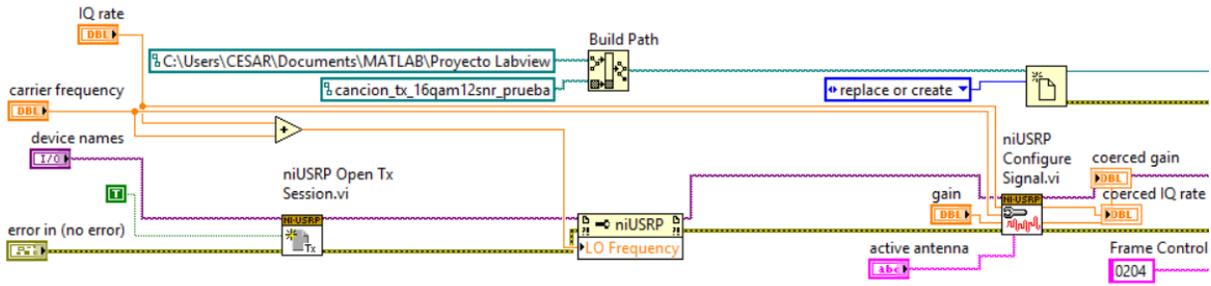


Figura 5.33 – Diagrama a bloques de inicio de conexión con el USRP e inicio de generación de archivo.

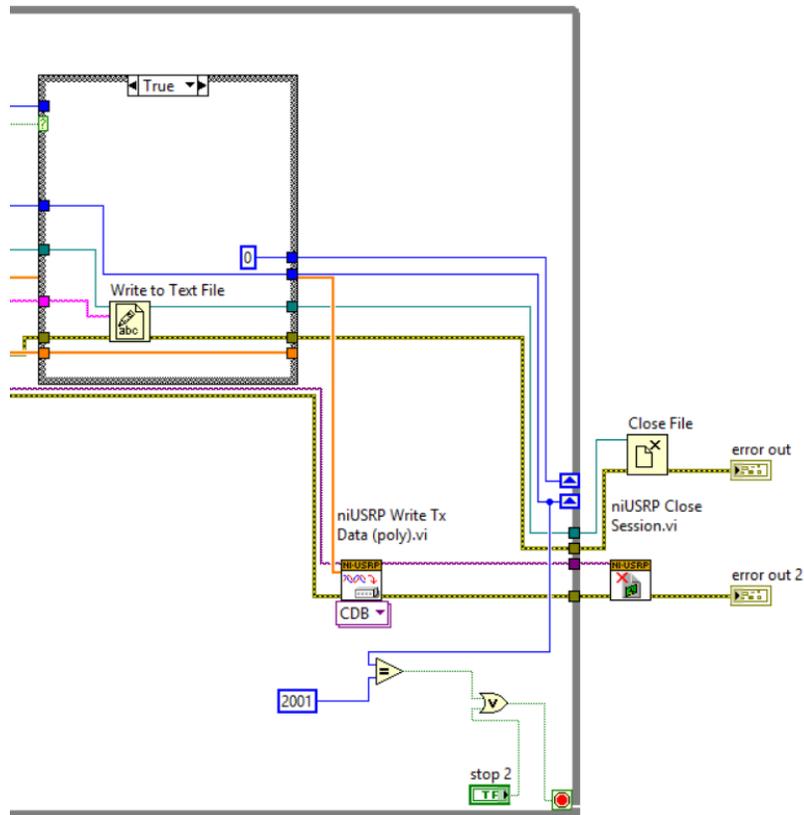


Figura 5.34 – Diagrama a bloques de conexión con el USRP y generación de archivo.

Es muy importante mencionar que todo el programa tiene que permanecer en un while infinito, para que los frames se envíen continuamente, además se agrega un contador donde se puede indicar el número de frames a enviar.

El programa que se encarga de analizar el BER se describe en el Anexo 1.

5.12 Sincronización

La etapa de sincronización es de suma importancia, ya que es la encargada, en la etapa de recepción, de analizar constantemente el canal hasta detectar las secuencias de entrenamiento transmitidas y así indicar que hay un frame y se pueda proceder con el procesamiento del frame y verificar la correcta recepción del frame.

A pesar de que la actual tesis describe la implementación de un transmisor IEEE 802.11, también se intervino en la implementación de la sincronización de la recepción. Dónde se tomó como referencia el trabajo de *Chia Horng-Liu* [8], tomando el primer método de sincronización corta, el cual se describe en la Figura 5.35.

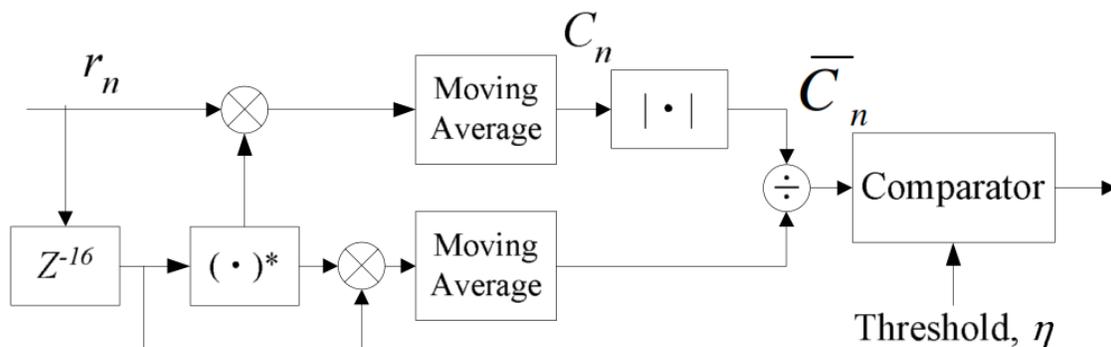


Figura 5.35 – Diagrama de detección de frame.

Éste método genera una ventana la cual va corriendo a través de los bits recibidos, verificando que el patrón de la secuencia corta enviada se repita tres veces. Para verificar la repetición de la secuencia corta se hace una correlación, en caso de que esto ocurra se dice que hay un frame y se envía a su posterior procesamiento. En LabVIEW se programa el bloque llamado *ShortSeq Detection2* (Figura 5.36), donde se

tienen dos secciones, la primera realiza el corrimiento de la ventana (Figura 5.37) y en la segunda se hace la correlación y se verifican las tres repeticiones (Figura 5.38).



Figura 5.36 – Bloque ShortSeq Detection2 de LabVIEW.

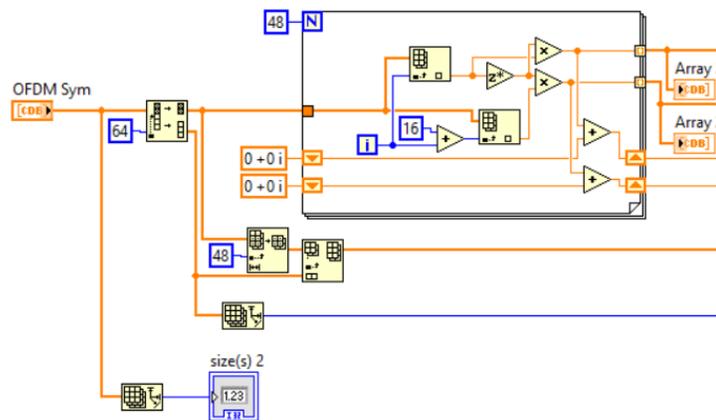


Figura 5.37 – Diagrama a bloques de corrimiento de ventana en la detección corta.

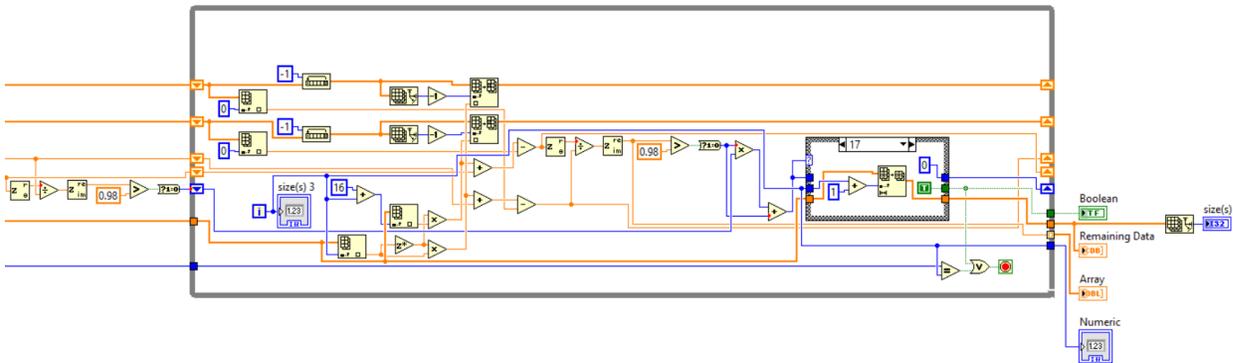


Figura 5.38 – Diagrama a bloques de correlación en la detección corta.

En éste capítulo se logró demostrar la implementación de un transmisor IEEE 802.11 en LabVIEW, donde se verificó el correcto funcionamiento de cada bloque de acuerdo al ejemplo descrito en el estándar IEEE 802.11.



Capítulo 6

Pruebas y Resultados

Introducción

En este trabajo de tesis se analiza el comportamiento, rendimiento y viabilidad de la implementación de la capa Física del estándar IEEE 802.11 en LabVIEW para SDR, como se mencionó anteriormente.

Para un total entendimiento del programa implementado. En el presente capítulo se mostraran distintas pruebas, las cuales midieron el BER en distintas condiciones y distancias utilizando los equipos USRP N210 y el programa hecho en LabVIEW de un transmisor IEEE 802.11g.

En las pruebas realizadas se utilizó un receptor, que también fue hecho en LabVIEW para la norma IEEE 802.11g, el cual es un trabajo de tesis distinto, pero es necesario para mostrar el rendimiento del transmisor y comprobar su funcionamiento.

Es muy importante mencionar que en todas las pruebas donde se obtuvo BER, fue mediante un programa que se realizó en MATLAB (Descrito en el Anexo) exclusivamente para los frames que generan en los programas de transmisión y recepción en LabVIEW.

Algo adicional que complementó y mejoró los resultados fue la investigación realizada en la San José State University, donde con ayuda del Dr. Robert Morelos Zaragoza se mejoró el código realizado, mejorando los resultados aquí presentados.

6.1 Descripción del Equipo Utilizado

El estándar al llevar tantas etapas de proceso de información, las cuales fueron descritas en el capítulo 5 de la presente tesis, la implementación en LabVIEW requiere una alta demanda de los recursos del sistema, por lo que se optó por usar computadoras con las características descritas a continuación.

Computadora para el transmisor: En la tabla 6.1 se pueden observar las características de procesamiento, memoria de acceso aleatorio (RAM), disco duro, tarjeta Ethernet, sistema operativo y aceleración de gráficos del equipo de cómputo que ejecutó el programa de transmisión.

Lenovo Y400	
Característica	Descripción
Procesador	3th Gen Intel Core i7-3630qm (2.4GHz 4MB)
Memoria RAM	8GB PC3-12800 DDR3L SDRAM 1600 MHz
Disco Duro	1 TB
Tarjeta Ethernet	Qualcomm Atheros AR8171/8175 PCI-E Gigabit Ethernet Card
Sistema Operativo	Windows 8.1 64
Acelerador de Gráficos	NVIDIA GeForce GT755M 2GB

Tabla 6.1 – Equipo de cómputo utilizado para el transmisor.

Computadora para el receptor: En la tabla 6.2 se pueden observar las características de procesamiento, memoria de acceso aleatorio (RAM), disco duro, tarjeta Ethernet, sistema operativo y aceleración de gráficos del equipo de cómputo que ejecutó el programa de transmisión.

Lenovo Y50	
Característica	Descripción
Procesador	4th Gen Intel Core i7-4720HQ Processor (2.5GHz 6MB)
Memoria RAM	8GB PC3-12800 DDR3L SDRAM 1600 MHz
Disco Duro	1 TB
Tarjeta Ethernet	Realtek RTL8111 Gigabit Ethernet Card
Sistema Operativo	Windows 8.1 64
Acelerador de Gráficos	NVIDIA GTX 860M 2GB (Maxwell)

Tabla 6.2 – Equipo de cómputo utilizado para el receptor.

SDR: El equipo SDR utilizado fue, tanto para transmisor como para receptor, el Ettus Research USRP N210 (Figura 6.1), con las tarjetas hijas SBX (Figura 6.2) de 400 MHz a 4400 MHz. Las características del USRP se muestran a continuación en la tabla 6.3.



Figura 6.1 – Ettus Research USRP N210.

Ettus Research USRP N210	
Característica	Descripción
FPGA	Xilinx Spartan 3A-DSP 3400
Convertidor A/D	Dual 100 MS/s
Convertidor D/A	Dual 400 MS/s
Interfaz	Gigabit Ethernet
Puertos	1 Tx/RX, 1 Rx, 1 MIMO, 1 Reloj, 1 Referencia, 1 GPS
Velocidad de procesamiento	50 MS/s

Tabla 6.3 – Características Ettus Research USRP N210.

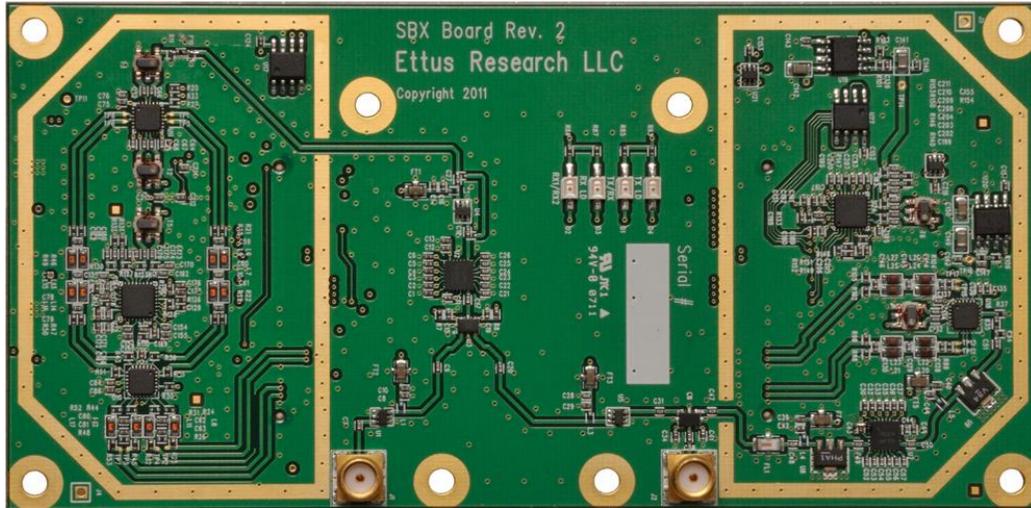


Figura 6.2 – Tarjeta hija SBX de 400 MHz a 4400 MHz.

Antenas: Se utilizaron dos tipos de antenas. Esto por el hecho de averiguar el comportamiento con una antena direccional y una omnidireccional. Los modelos utilizados fueron la LP0965 Yagi PCB (Figura 6.3) de 850 MHz a 6.5 GHz con ganancia de 5 dBi y la VERT 2450 (Figura 6.4) de banda dual de 2.4 a 2.48 GHz con ganancia de 3dBi.



Figura 6.3 – Antena LP0965 de 850 MHz a 6.5 GHz.



Figura 6.4 – Antena VERT2450 de 2.4 GHz a 2.48 GHz.

6.2 Obtención de la Frecuencia a Transmitir

Antes de hacer cualquier prueba del sistema realizado, se hizo un análisis del espectro para encontrar la frecuencia ideal para realizar la transmisión y recepción de datos, ya que Wi-Fi, que usa la banda de 2.4 GHz, provoca que existan muchas interferencias en el rango de frecuencias donde opera, como se puede observar en las Figuras 6.5 y 6.6.

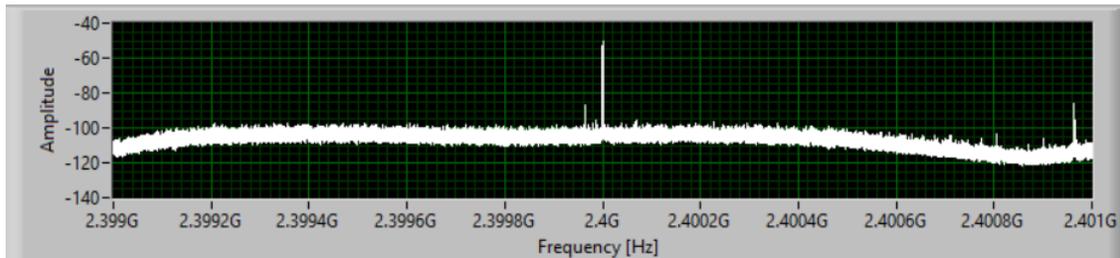


Figura 6.5 – Espectro con frecuencia central en 2.4 GHz.

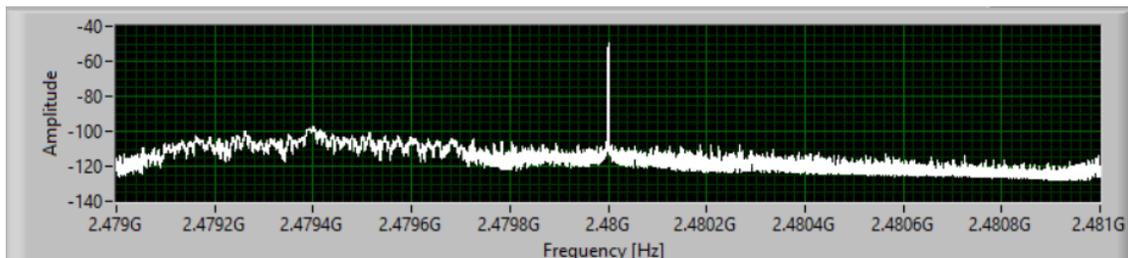


Figura 6.6 – Espectro con frecuencia central en 2.48 GHz.

Como se puede observar el espectro de Wi-Fi tiene muchas interferencias, por lo que se decidió buscar en una frecuencia cercana a 2.4 GHz, encontrando un espacio libre en 2.6 GHz, como se puede observar en la Figura 6.7.

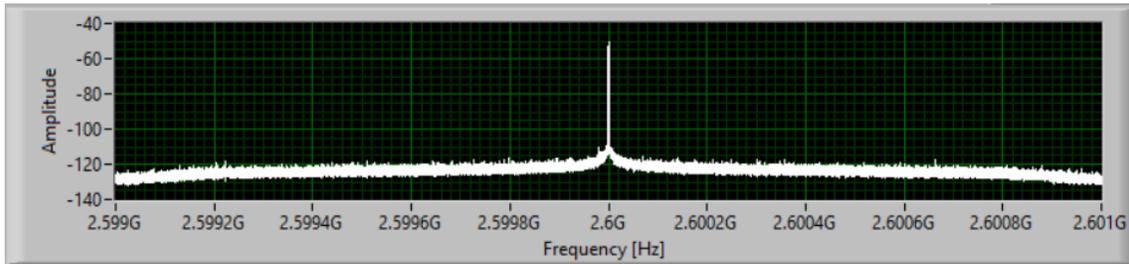


Figura 6.7 – Espectro con frecuencia central en 2.6 GHz.

Gracias al análisis espectral realizado, se decidió utilizar la banda de 2.6 GHz para todas las pruebas de transmisión que se explicarán adelante.

6.3 Prueba de sincronía y constelaciones

La primera prueba realizada fue comprobar la sincronía implementada entre el transmisor y el receptor, las pruebas se realizaron en un laboratorio donde las interferencias se redujeron (Figura 6.8), a 1 metro de distancia, 12 dB de ganancia de transmisión y 2 dB de ganancia de recepción, utilizando las antenas LP0965.



Figura 6.8 – Escenario para la prueba de sincronía y constelaciones.

Para realizar las pruebas, se transmitió el frame propuesto en el estándar IEEE 802.11 y se incluyó la Figura 6.9 en LabVIEW la cual mostró la correlación en la sincronización. Además para tener una mayor percepción de los frames recibidos se agregó un indicador booleano (Figura 6.10) (Funciona como un LED), prendiendo cuando un frame era detectado y se incluyó un contador de frames detectados (Figura 6.10).

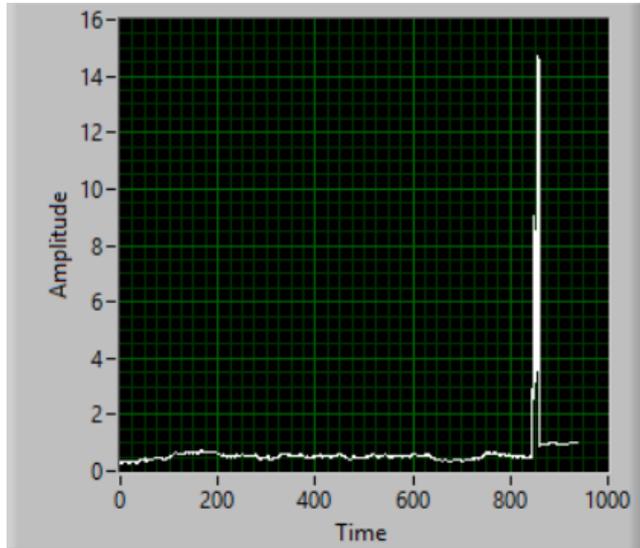


Figura 6.9 – Correlación obtenida en la etapa de Sincronización.

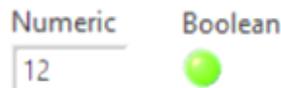


Figura 6.10 – Indicadores que muestran la detección de un frame.

Cómo se puede observar en la Figura 6.10, el algoritmo implementado funciona correctamente, ya que la cantidad de frame detectados es igual a los frames que se enviaron, además la Figura 6.9 confirma que existe un frame, ya que hay un pico el cual indica una correlación.

Una vez que se comprobó la correcta recepción de frames, se procedió a mostrar las constelaciones de las modulaciones incluidas. Cabe destacar que se comprobó que todas las modulaciones funcionan correctamente, pero con fines demostrativos en la presente tesis solo se incluirán pruebas para QPSK 1/2 y para 64-QAM 1/2.

En la Figura 6.11 se puede observar la constelación QPSK 1/2 transmitida y en la Figura 6.12 se observa la constelación 64-QAM 1/2 transmitida. Las constelaciones recibidas se pueden observar en la Figura 6.13 y la Figura 6.14.

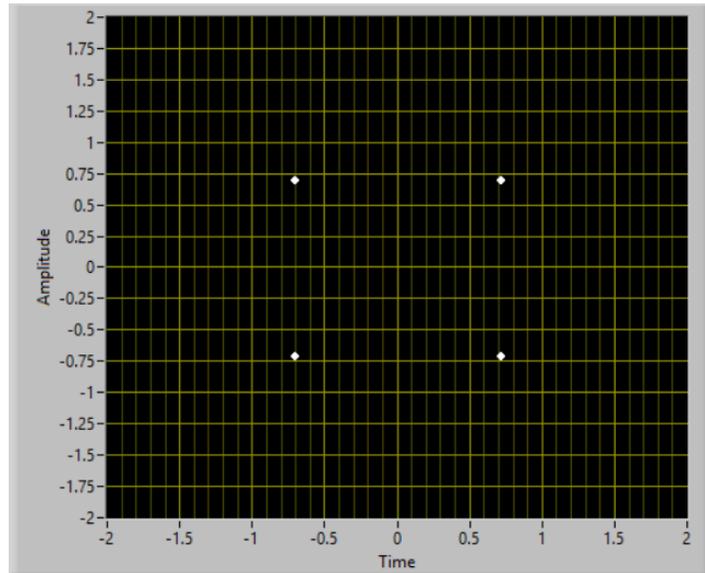


Figura 6.11 – Constelación QPSK 1/2 transmitida.

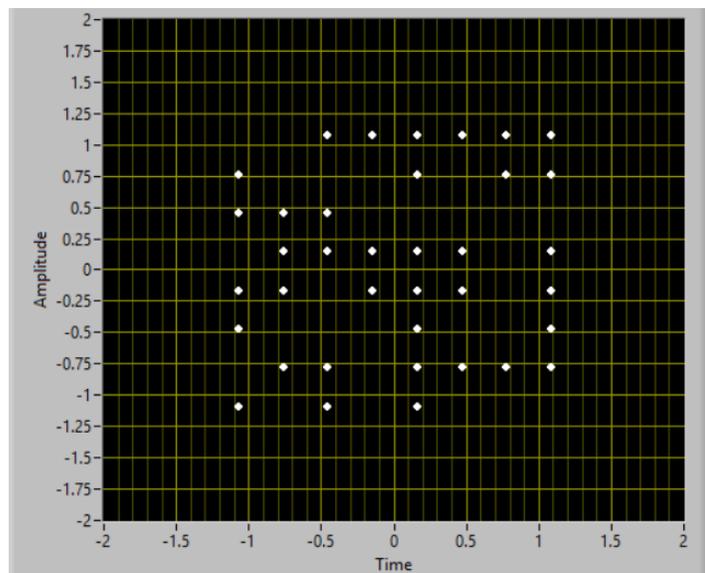


Figura 6.12 – Constelación 64-QAM 1/2 transmitida.

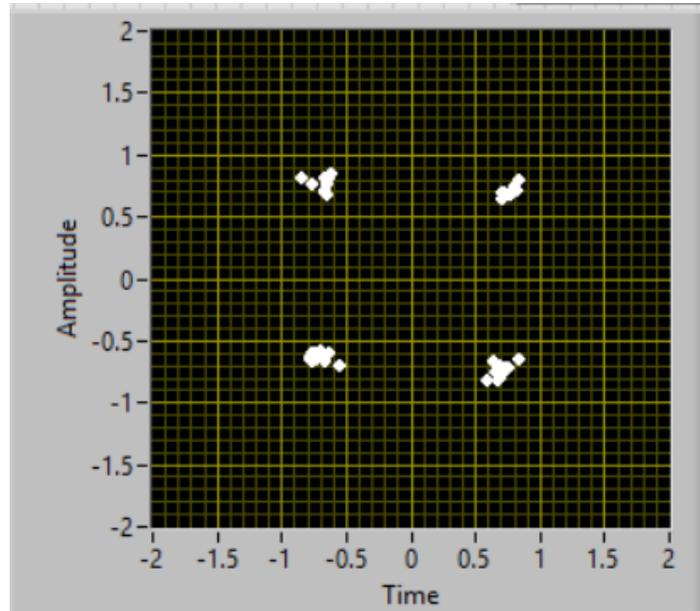


Figura 6.13 – Constelación QPSK 1/2 recibida.

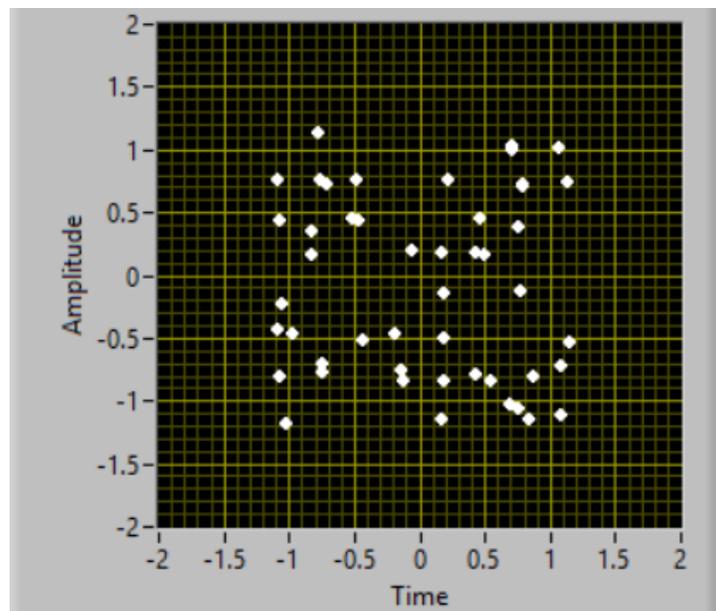


Figura 6.14 – Constelación 64-QAM 1/2 recibida.

Como se pudo observar en las Figuras 6.13 y 6.14 las constelaciones recibidas se definen, a pesar de los efectos de multitrayectorias, atenuaciones e interferencias, es posible interpretar los valores recibidos y mapearlos a bits correctamente.

Al comparar las constelaciones de transmisión y recepción se pudo confirmar que el transmisor funciona correctamente, ya que el procesamiento realizado logró ser interpretado por el receptor IEEE 802.11.

6.4 Medición del Espectro

La siguiente prueba realizada, fue el análisis del espectro generado por el transmisor, con el fin de observar el símbolo OFDM generado, la prueba se realizó en el laboratorio de redes inalámbricas del edificio Q del Anexo de Ingeniería (Figura 6.8), a 1 metro de distancia, 12 dB de ganancia de transmisión y 2 dB de ganancia de recepción.

Para realizar la prueba se utilizó el ejemplo de LabVIEW “*niUSRP EX Spectral Monitoring (Interactive)*”, con el cual se analizó la banda de 2.6 GHz como se muestra en la Figura 6.15. Debido al alto procesamiento requerido y la gran cantidad de datos a analizar, el sistema operativo de Windows otorga un buffer muy pequeño al momento de usar el programa del receptor IEEE 802.11, lo que afecta directamente al ancho de banda llegando solo utilizar hasta 2 MHz de los 20 MHz que pide el estándar.

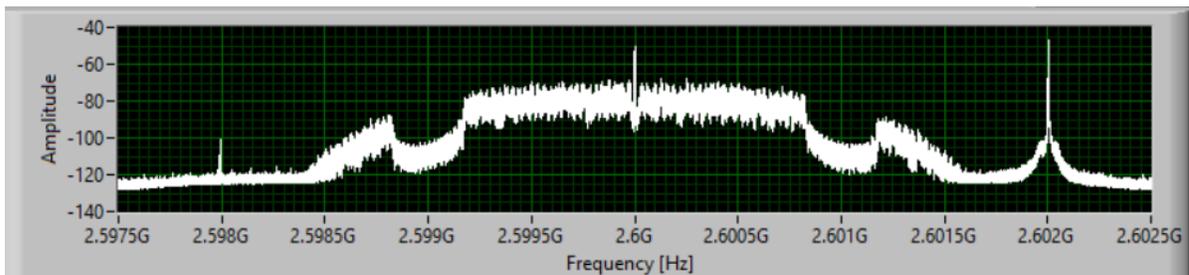


Figura 6.15 – Espectro de la señal transmitida.

Como se puede observar en la Figura 6.15 el ancho de banda de la señal es de 2 MHz, confirmando que la señal enviada es correcta, pero surgen picos en las laterales del símbolo OFDM. Para solucionar esto, se utilizó un multiplicador en el programa de

transmisión de LabVIEW antes del bloque *niUSRP Write Tx Data (poly) VI* y se volvió a analizar el espectro, como se muestra en la Figura 6.16.

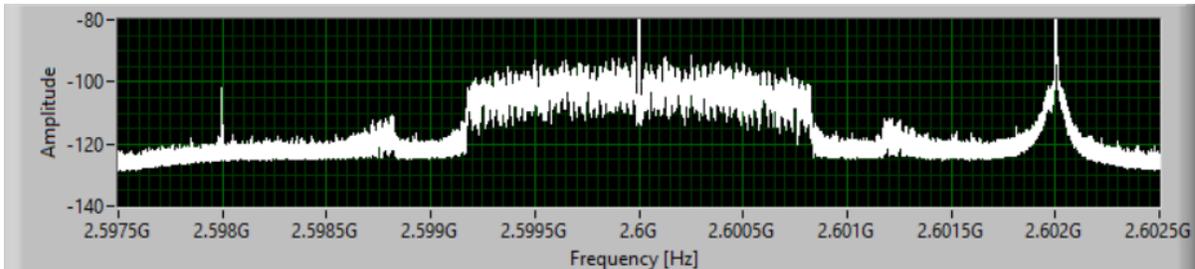


Figura 6.16 – Espectro de la señal transmitida con multiplicador.

Al enviar el valor 1 o muy cercano a 1, los amplificadores del USRP N210 se saturan deformando la señal transmitida, por eso es muy importante en algunos casos atenuar la señal, para que ésta se comporte correctamente y como se puede observar en la Figura 6.16 el problema de los picos en las laterales del símbolo OFDM se solucionó utilizando una constante de 0.5 para QPSK y de 0.125 para 64-QAM.

6.5 Prueba de Bit Error Rate a 1m de Distancia

Una de las pruebas para conocer la fiabilidad de un sistema, es la medición del BER, ya que esta nos indica cuantos bits en error llegan por cada n número de bits enviados. Para realizar ésta prueba se tiene que tener un control muy exacto de los bits que se envían para que se pueda hacer una comparación bit a bit al momento de recibirlos. Para la prueba se utilizó el escenario de la Figura 6.8 y se envió el mensaje que se da de ejemplo en el estándar IEEE 802.11, el cual se encuentra a continuación.

Joy, bright spark of divinity,

Daughter of Elysium,

Fire-insired we trea

Éste mensaje consta de 100 bytes u 800 bits. Se decidió enviar el mensaje cada 200 ms para asegurar que los frames se reciban correctamente, ya que el receptor no



puede interpretar los mensajes si se envían cada 50 ms o menos. También se utilizó un ancho de banda de 2 MHz (máximo alcanzado) con frecuencia central en 2.6 GHz.

Para calcular la velocidad de transmisión de lo que se está enviando se utiliza la Fórmula 6-1.

$$\text{Velocidad de transmisión} = \frac{1}{\text{Duración del Símbolo}} \times cc \times m \times N_{FFT} \quad (6-1)$$

Dónde

cc= tasa de codificación

m = Bits por punto de la constelación

N_{FFT} = Portadoras de Datos

Sustituyendo en 6-1 para QPSK 1/2 donde m=2

$$\text{Velocidad de transmisión} = \frac{1}{40 \text{ ms}} \times \frac{1}{2} \times 2 \times 48$$

$$\text{Velocidad de transmisión} = 1.2 \text{ Mb/s}$$

Sustituyendo para 64-QAM 1/2 donde m=6

$$\text{Velocidad de transmisión} = \frac{1}{40 \text{ ms}} \times \frac{1}{2} \times 6 \times 48$$

$$\text{Velocidad de transmisión} = 3.6 \text{ Mb/s}$$

Se puede observar que las velocidades de transmisión son 1.2 Mb/s y 3.6 Mb/s para QPSK 1/2 y 64-QAM 1/2 respectivamente.

La antena utilizada fue la LP0965, ya que al ser muy direccional y de alta ganancia aseguró que la potencia fuera aprovechada al máximo, obteniéndose así los mejores resultados posibles, de acuerdo al equipo con el que se contaba.

La prueba de BER se realizó variando la ganancia de transmisión y midiendo el SNR (Tabla 6.4 y en la Tabla 6.5) para cada punto de la medición, donde el ruido era



constante en -121 dBm. Los resultados de la prueba se pueden observar en la Figura 6.17 para QPSK 1/2 y en la Figura 6.18 para 64-QAM 1/2.

QPSK 1/2, AMP=0.5	
SNR	BER
6	0.7808
7.7	0.1809
9.1	0.0013
11.6	6.60E-04
14.4	6.60E-04
17.1	9.90E-04
18.9	3.30E-04
24.1	0.00E+00

Tabla 6.4 – SNR vs BER medida para QPSK 1/2.

64-QAM 1/2, AMP=0.125	
SNR	BER
13.7	0.4582
14.9	0.4511
16.3	0.2498
18.2	1.00E-02
20.01	1.90E-03
21.6	3.27E-04

Tabla 6.5 – SNR vs BER medida para 64-QAM 1/2.

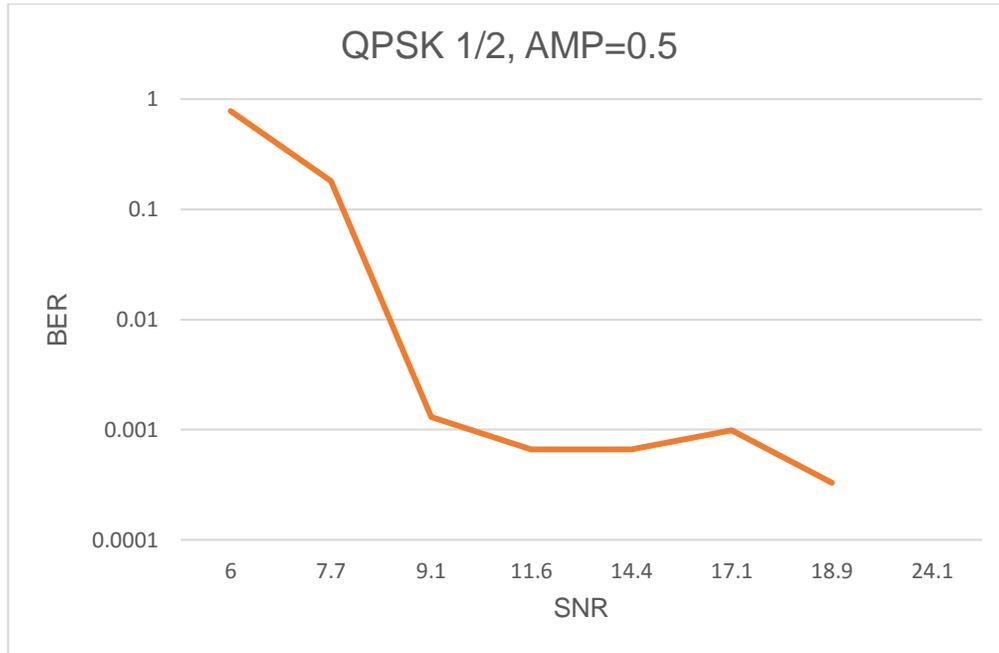


Figura 6.17 – SNR vs BER para QPSK 1/2.

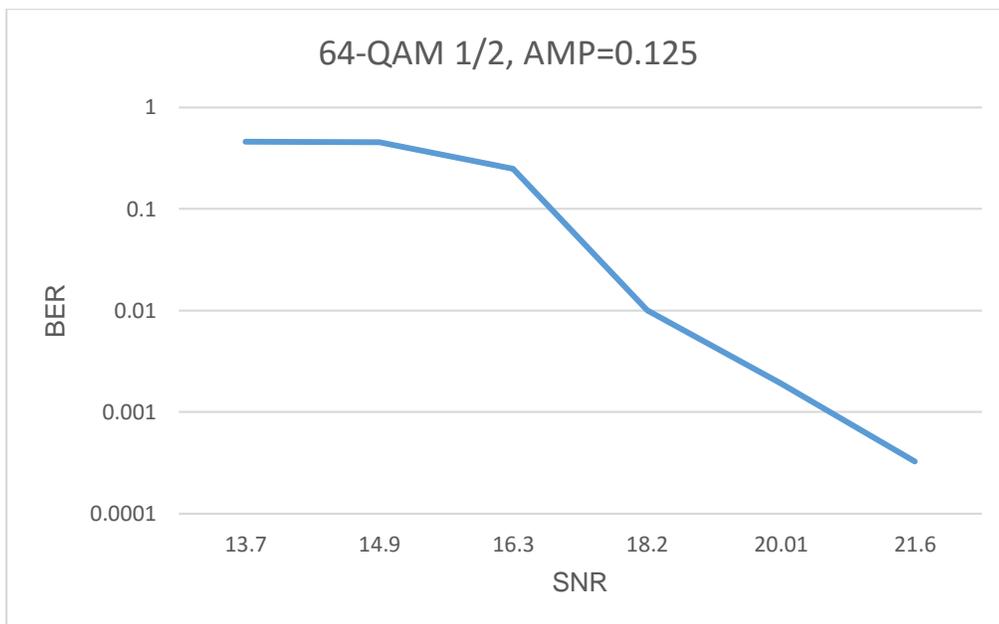


Figura 6.18 – SNR vs BER para 64-QAM 1/2.

Como se puede observar en los resultados obtenidos mientras más compleja sea la modulación el BER incrementa, pero la velocidad de transmisión aumenta. Los menores BER obtenidos fueron alrededor de los 20 dB para las dos modulaciones, siendo 0 para QPSK y 3.27×10^{-4} para 64-QAM.

6.6 Prueba de Distancia contra BER

La última prueba realizada se trató de enviar el mismo frame, con las mismas características del canal y velocidad de transmisión, que se describieron en el subtema 6.4, pero variando la distancia en interiores y exteriores, tanto para la antena LP0965, como para VERT2450.

6.6.1 Distancia Vs BER en interiores

Estas pruebas fueron realizadas en un pasillo del edificio 'Q' del Anexo de Ingeniería de la Universidad Nacional Autónoma de México (Figura 6.19), ya que presentaba las características habituales de un ambiente en interiores. Se empezó por dejar fijo el receptor e ir recorriendo el transmisor cada metro para la antena LP0965 y cada dos 2 metros para la antena VERT2450, como se puede observar en las Tablas 6.4 y 6.5 para QPSK 1/2 en las Tablas 6.6 y 6.7 para 64-QAM 1/2.



Figura 6.19 – Escenario para la prueba de distancia en interiores.

QPSK 1/2	
Distancia[m]	BER
1	0
2	9.99E-04
3	0.0025
4	0.004
5	0.0085
6	0.0065
7	0.0042
8	0.0045
9	0.0393
10	0.0387
11	0.1612

Tabla 6.5 – Distancia vs BER para QPSK 1/2 utilizando la antena LP0965 en interiores.

QPSK 1/2	
Distancia[m]	BER
2	0.3397
4	0.2392
6	0.2294
8	0.4904

Tabla 6.6 – Distancia vs BER para QPSK 1/2 utilizando la antena VERT2450 en interiores.

64-QAM 1/2	
Distancia[m]	BER
1	7.70E-04
2	0.009
3	0.0096
4	0.0055
5	0.0041
6	0.0094
7	0.0281
8	0.0053
9	0.1532
10	0.1558
11	0.2861

Tabla 6.7 – Distancia vs BER para 64-QAM 1/2 utilizando la antena LP0965 en interiores.

64-QAM 1/2	
Distancia[m]	BER
2	0.191
4	0.2681
6	0.4783
8	0.4367

Tabla 6.8 – Distancia vs BER para 64-QAM 1/2 utilizando la antena VERT2450 en interiores.

Después de obtener los resultados se procedió a realizar las gráficas con MATLAB, de manera que se pudiera comparar el BER de acuerdo la modulación en cada antena, como se puede observar en la Figura 6.20 para la antena LP0965 y en la Figura 6.21 para la antena VERT2450.

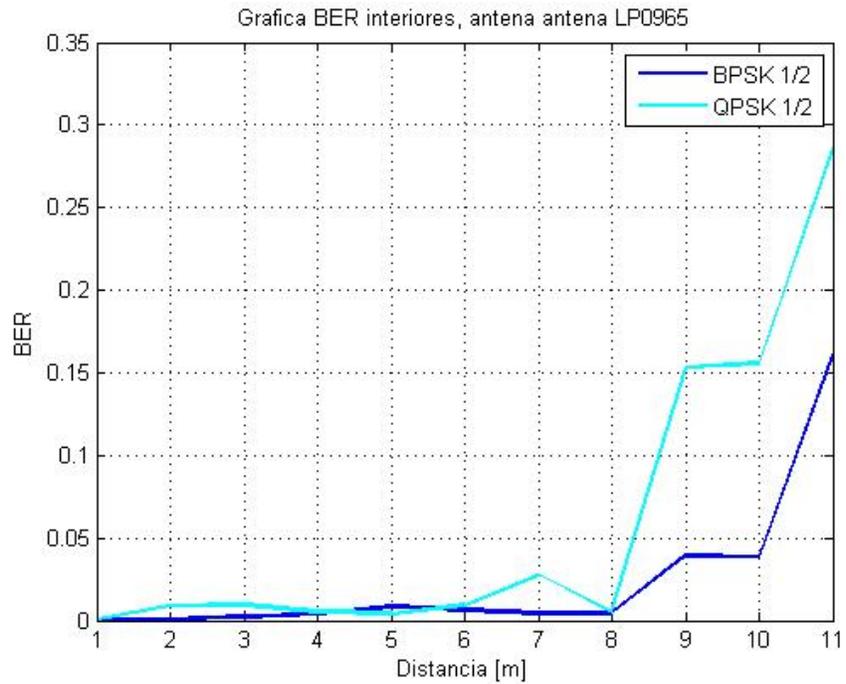


Figura 6.20 – Distancia vs BER para la antena LP0965 en interiores.

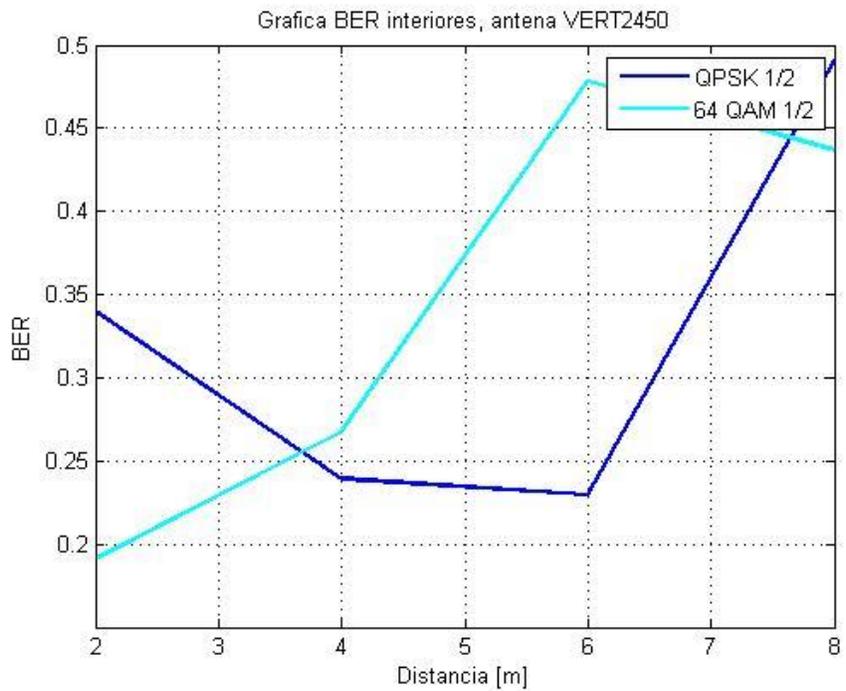


Figura 6.21 – Distancia vs BER para la antena VERT2450 en interiores.

Como se puede observar la antena VERT2450 no funciona bien para esta aplicación ya que 2.6 GHz es una frecuencia muy alta y por limitantes de equipo no se pudo utilizar otra antena omnidireccional. Tampoco se cambió la frecuencia de operación ya que las frecuencias en donde trabaja correctamente la antena VERT2450 existen muchas redes Wi-Fi lo que afecta las mediciones.

La antena LP065 funciona de manera correcta en interiores, ya que si se observa la Figura 6.20, tiene un BER que va de 0 hasta 10^{-3} para QPSK y de 7.7×10^{-4} hasta 0.2 para 64-QAM. Los resultados son satisfactorios ya que a pesar de ser pruebas experimentales para equipos que no habían sido utilizados en este tipo de implementaciones, el BER es muy bajo, hasta para una modulación como es 64-QAM.

6.6.2 Distancia Vs BER en exteriores

La prueba en exteriores se realizó en la azotea del edificio 'Q' del Anexo de Ingeniería de la Universidad Nacional Autónoma de México (Figura 6.22), donde se tuvo un ambiente libre de obstáculo minimizando las reflexiones ya que no había paredes ni otros objetos que pudieran ocasionar demasiados frentes de onda.



Figura 6.22 – Escenario para la prueba de distancia en exteriores.



Las mediciones se realizaron prácticamente igual que en interiores, pero esta vez se observó que se alcanzaba una distancia mucho mayor, la cual fue de 21.5m, así que se tomó este como punto de referencia y se tomaron cuatro puntos distintos para cada modulación como se puede observar en la Tabla 6.9 y en la Tabla 6.10. En esta ocasión no se presentaron resultados para la antena VERT2450 ya que no se podía capturar la suficiente cantidad de frame correctos para obtener un BER, a pesar de posicionar los equipos a unos cuantos centímetros de distancia, así que todas las mediciones corresponden para la antena LP0965.

QPSK 1/2	
Distancia[m]	BER
2	0.0035
9	0.0195
16	0.0125
21	0.1309

Tabla 6.9 – Distancia vs BER para QPSK 1/2 utilizando la antena LP0965 en exteriores.

64-QAM 1/2	
Distancia[m]	BER
2	0.0055
9	0.0286
16	0.038
21	0.1809

Tabla 6.10 – Distancia vs BER para 64-QAM 1/2 utilizando la antena LP0965 en exteriores.

Para observar una comparación entre las modulaciones en base a la distancia se graficaron los resultados con ayuda de MATLAB. Como se puede observar en la Figura 6.23.

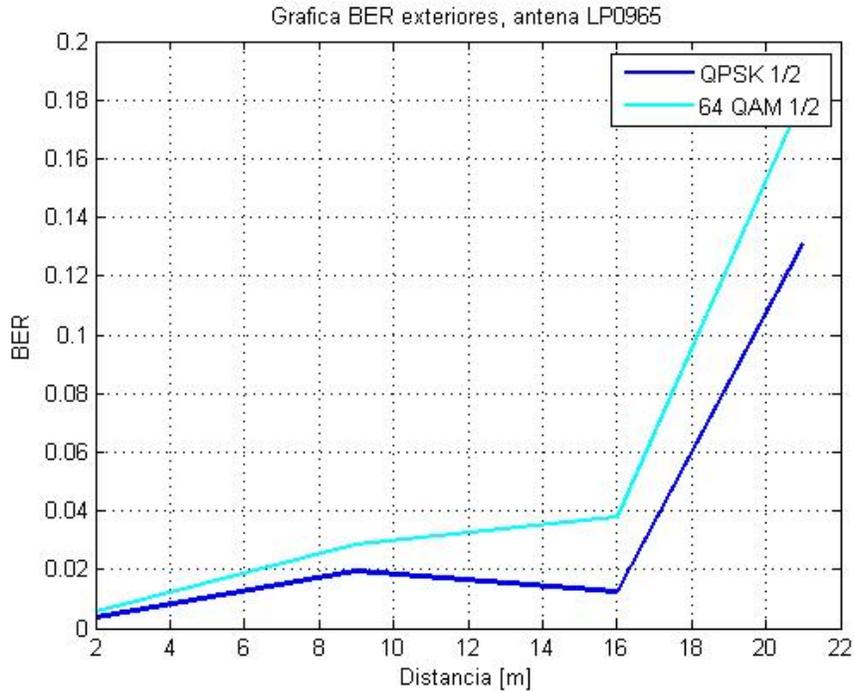


Figura 6.23 – Distancia vs BER para la antena LP0965 en exteriores.

En la prueba de exteriores se pueden observar grandes cambios respecto a interiores. El primer cambio fue que el BER aumento de manera considerable ya que el mínimo obtenido fue alrededor de 10^{-3} , para las dos modulaciones, el segundo cambio es la distancia máxima ya que esta aumento considerablemente logrando transmitir mensajes hasta una distancia de 21.5m. Por último el comportamiento del BER es más estable ya que se mantiene prácticamente igual hasta los 16m donde se observa un aumento abrupto en los valores medidos.



Conclusiones

En la presente tesis se implementó un transmisor IEEE 802.11g en LabVIEW, utilizando los equipos USRP N210 para la transmisión, de los datos generados, por un medio físico y realizando pruebas de BER en distintos entornos para conocer la viabilidad del sistema para su uso comercial.

Primeramente la programación realizada en LabVIEW, la cual se basó en el estándar IEEE 802.11g, fue exitosa, ya que al transmitir el frame que viene como ejemplo en el estándar, se logró volver a obtener los datos enviados en la recepción, confirmando que el procesamiento realizado en la transmisión fue el correcto y cumpliendo así las especificaciones de procesamiento en banda base de capa PHY que indica la norma IEEE 802.11g.

Al existir una limitante en los equipos USRP N210 en la capacidad de la interfaz que va hacia un ordenador, el ancho de banda (I/Q Rate) se ve reducido en la recepción, por lo que el transmisor a pesar de alcanzar los 20 MHz propuestos en el estándar tiene que trabajar con 2 MHz como máximo de ancho de banda, lo que no es un problema, ya que aun así se logró una correcta implementación de un transmisor IEEE 802.11g, por lo que se puede concluir que el objetivo principal de la tesis se cumplió.

Para verificar la viabilidad del sistema para su uso comercial, se realizaron distintas pruebas donde se midió el BER, en diferentes entornos, con diferentes antenas y a diferentes distancias, lo que entregó una gran cantidad de datos, los cuales muestran que tanto en exteriores como en interiores a una distancia máxima de 16 m se puede obtener un BER aceptable, donde máximo se logró obtener 9.99×10^{-4} de 2 m - 4 m de distancia y 0 a 1 m de distancia, con modulación QPSK 1/2.

También es muy importante mencionar que la modulación 64-QAM 1/2 es la primera vez que se logra utilizar correctamente en un transmisor IEEE 802.11g utilizando SDR, ya que la programación realizada permitió obtener un BER 9.99×10^{-4} a 1 m de



distancia, lo que permite concluir que el transmisor IEEE 802.11g que se implementó, cumple con características que pueden utilizarse en aplicaciones donde se requiera enviar una cantidad de información pequeña, como avisos y en donde la velocidad de transmisión no sea un factor crítico, ya que la reducción de ancho de banda que se tuvo que realizar permitió obtener como máximo 3.6 Mb/s con 64-QAM, por lo que se puede concluir que el segundo objetivo se cumple siempre y cuando se utilice para aplicaciones que requieran tasas de transmisión de hasta 3.6 Mb/s en caso de utilizar el receptor que se programó en la tesis de receptor IEEE 802.11g.

Por último es importante concluir que los algoritmos de sincronización se pueden implementar en equipos SDR y conseguir una correcta detección de frame, como se mostró en el capítulo de resultados de la presente tesis, aunque su aplicación requiere una gran cantidad de procesamiento.

Existen distintas cosas que se podrían mejorar, como el ancho de banda, si se aplican los programas en otros equipos o se implementa una capa MAC, como se mostrará en el trabajo a futuro.



Trabajo a Futuro

Como en todo sistema existen cosas que se pueden mejorar, pero la implementación de estas mejoras requiere otro gran lapso de tiempo para que se pueda confirmar que efectivamente benefician al funcionamiento del sistema.

Una de las principales mejoras que se pueden realizar a futuro es utilizar otro equipo llamado USRP E300, en lugar del USRP N210. El USRP E300 tiene la peculiaridad de tener un ordenador integrado, lo que eliminaría el problema del búfer de dato de la interfaz y así se podría verificar si el sistema en general logra recibir información con un ancho de banda de 10 MHz o 20 MHz, el problema es que el sistema operativo que contiene el USRP E300 es Linux, por lo que habría que migrar el código realizado en LabVIEW a GNU Radio.

Otro trabajo a futuro que se puede realizar a la tesis es la adición de una capa MAC, la cual pueda verificar mediante ACKs el envío y recepción de paquetes, realizando retransmisiones en caso de necesitarse, esto limitaría el throughput máximo, pero lograría que los datos siempre lleguen correctamente al destino, permitiendo que el sistema se pueda utilizar en aplicaciones que requieran una mayor fiabilidad en el envío y recepción de la información.

En caso de realizar las mejoras antes mencionados permitiría que el transmisor y receptor IEEE 802.11g logren utilizarse en su capacidad máxima y se podría verificar su uso comercial en y comparar con todas las aplicaciones de transmisión y recepción IEEE 802.11 de circuitería convencional.



Anexo 1

Análisis de BER

Para realizar en análisis de BER se realizó un programa en MatLab llamado *BER/v*, el cual lee el archivo que LabVIEW genera en el transmisor, reconociendo mediante el patrón decimal '10 10 64 64 10 10', la existencia de un frame y lo acomoda en orden en un nuevo arreglo, este paso se englobó en la función con nombre *Transmisión*, el código se presenta a continuación.

```
function [ b_vec ] = Transmision(Nombre_tx)

a=fopen(Nombre_tx,'r');
b=fread(a);
c=6;h=1;
newline=find(b==10);
nfr=0;

for j=2:length(newline)-1
    if newline(j)-newline(j-1)==1
        if newline(j+1)-newline(j)==3
            if newline(j+2)-newline(j+1)==1
                if (b(newline(j)+1)==64)
                    if (b(newline(j)+2)==64)
                        nfr=nfr+1;
                    end
                end
            end
        end
    end
end
startline=zeros(nfr,1);

for j=2:length(newline)-1
    if newline(j)-newline(j-1)==1
        if newline(j+1)-newline(j)==3
            if newline(j+2)-newline(j+1)==1
                if (b(newline(j)+1)==64)
                    if (b(newline(j)+2)==64)
                        startline(h)=newline(j+2);
                        h=h+1;
                    end
                end
            end
        end
    end
end
```



```
end
end
end
end
end

d=length(startline);e=startline(2)-12;
b_vec=zeros(e,d);

for j=1:d
    for k=1:e
        c=c+1;
        b_vec(k,j)=b(c);

        end
        %b_vec(e+1,j)=12;
        c=c+6;
    end

end

end
```

Después de tener los frames transmisión en orden se realiza una comparación con los frames obtenidos en la recepción, pero al existir errores en los frames recibidos, algunas veces los números de secuencia se pierden por lo que no se puede ordenar algunos frames. Se decidió en base a probabilidades buscar frames que coincidieran al menos un 30% con algún frame transmitido. Esta decisión se basa en la longitud del frame y en las correspondencias que se obtengan con las posiciones comparadas.

Al rescatar los frames sin secuencia a los que se les llamo basura, se ordenan los frames detectados y los frames basura en un arreglo y así se realiza la comparación bit a bit de los frames transmitidos y los frames recibidos. El código de extracción de basura y comparación de bits se muestra a continuación.

```
b_rx=b_frame;
```



```
b_rx2=zeros(size(b_tx));
WrBit=0;
n_Framenrx=0;
n_WrFrame=0;
n1=size(b_tx);
n2=size(b_rx);
%
%b_rx2(1:n1(1),1:n1(2))=b_rx(1:n1(1),1:n1(2));
%b_rx=b_rx2;
%

b_tx2=zeros(max(n1(1),n2(1)),max(n1(2),n2(2)));
b_rx2=zeros(max(n1(1),n2(1)),max(n1(2),n2(2)));
for j=1:n1(2)
    for k=1:n1(1)
        b_tx2(k,j)=b_tx(k,j);
    end
end
for j=1:n2(2)
    for k=1:n2(1)
        b_rx2(k,j)=b_rx(k,j);
    end
end
end

b_tx=b_tx2;
b_rx=b_rx2;

nfr=zeros(1,2);
nfr(1)=max(n1(1),n2(1));
nfr(2)=max(n1(2),n2(2));

Wr_Frame=zeros(1,nfr(2));
NRx_Frame=zeros(1,nfr(2));
n3=size(garbage);
%garbage=[garbage ; zeros(1,n3(2))];
for j=1:nfr(2)
    if (b_rx(:,j)==0)
        WrBit=WrBit+nfr(1)*8;
        n_Framenrx=n_Framenrx+1;
        NRx_Frame(j)=1;
    else
        diff=find((b_tx(:,j)-b_rx(:,j))~=0);
        if length(diff)~=0 %#ok<ISMT>
            n_WrFrame=n_WrFrame+1;
            Wr_Frame(j)=1;
        end
        for k=1:length(diff)
            m=dec2bin(b_tx(diff(k),j));
            n=dec2bin(b_rx(diff(k),j));
            while length(m)<8,
                m=strcat('0',m);
            end
            if length(m)>8
                A=regexp(m, '\w{1,8}', 'match');
```



```
        m=char(A(1));
    end

    while length(n)<8,
        n=strcat('0',n);
    end
    if length(n)>8
        A=regexp(n, '\w{1,8}', 'match');
        n=char(A(1));
    end
    WrBit=WrBit+length(find((m-n)~=0));
end
end

BER=WrBit/(nfr(1)*nfr(2)*8);
Wr_Frame=find(Wr_Frame==1);
NRx_Frame=find(NRx_Frame);

g=size(garbage);
h=1;
% rel=zeros(g(2),length(NRx_Frame));
%
% for j=1:g(2)
%     for h=1:length(NRx_Frame)
%         for k=1:n1(1)
%             m=dec2bin(garbage(k,j));
%             n=dec2bin(b_tx(k,NRx_Frame(h)));
%             while length(m)~=8, m=strcat('0',m); end
%             while length(n)~=8, n=strcat('0',n); end
%             rel(j,h)=rel(j,h)+length(find((m-n)~=0));
%         end
%     end
% end

ctx=size(b_tx);
comp=zeros(length(gar_pos),length(NRx_Frame));

for j=1:1:length(gar_pos)
    i=0;
    for k=1:1:length(NRx_Frame)
        l=NRx_Frame(k);
        i=1+i;
        for h=1:1:n1(1)
            if garbage(h,j)==b_tx(h,l)
                comp(j,k)=comp(j,k)+1;
            end
        end
    end
end

end
threshold=0.3*nfr(1);
mej=zeros(length(gar_pos));
for j=1:length(gar_pos)
```



```
    if (length(find(comp(j, :)==max(comp(j, :)) &max(comp(j, :))>threshold))==1)
        mej(j)=find(comp(j, :)==max(comp(j, :)) &max(comp(j, :))>threshold);
    elseif
        (length(find(comp(j, :)==max(comp(j, :)) &max(comp(j, :))>threshold))>1)
        t=find(comp(j, :)==max(comp(j, :)) &max(comp(j, :))>threshold);
        mej(j)=t(1);
    else
        mej(j)=0;
    end
end

end

%mej=mej(find(mej~=0));
mej=mej(:,1);
for j=1:length(mej)
    if mej(j)~=0
        b_rx(:,NRx_Frame(mej(j)))=garbage(1:n1(1),j);
    end
end

b_rx2=zeros(size(b_tx));
WrBit=0;
n_Framenrx=0;
n_WrFrame=0;
Wr_Frame=0;
NRx_Frame=0;

for j=1:nfr(2)
    if (b_rx(:,j)==0)
        WrBit=WrBit+nfr(1)*8;
        n_Framenrx=n_Framenrx+1;
        NRx_Frame(j)=1;
    else
        diff=find((b_tx(:,j)-b_rx(:,j))~=0);
        if length(diff)~=0 %#ok<ISMT>
            n_WrFrame=n_WrFrame+1;
            Wr_Frame(j)=1;
        end
        for k=1:length(diff)
            m=dec2bin(abs(b_tx(diff(k),j)));
            n=dec2bin(abs(b_rx(diff(k),j)));
            while length(m)<8,
                m=strcat('0',m);
            end
            if length(m)>8
                A=regexp(m, '\w{1,8}', 'match');
                m=char(A(1));
            end

            while length(n)<8,
                n=strcat('0',n);
            end
            if length(n)>8
```



```
        A=regexp(n, '\w{1,8}', 'match');
        n=char(A(1));
    end
    WrBit=WrBit+length(find((m-n)~=0));
end
end
end
BER=WrBit/(nfr(1)*nfr(2)*8);
Wr_Frame=find(Wr_Frame==1);
NRx_Frame=find(NRx_Frame);
```



Lista de Términos

16-QAM	<i>16 Quadrature Amplitude Modulation</i>
64-QAM	<i>64 Quadrature Amplitude Modulation</i>
ACK	<i>Acknowledgment</i>
ADC	<i>Analog Digital Converter</i>
AIFS	<i>Arbitrary Interframe Space</i>
BER	<i>Bit Error Rate</i>
BPSK	<i>Binary Phase Shift Keying</i>
BSS	<i>Basic Service Set</i>
CCK	<i>Complementary Code Keying</i>
CP	<i>Cyclic Prefix</i>
CPU	<i>Central Processing Unit</i>
CRC	<i>Cyclic Redundancy Check</i>
CSMA/CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
CSMA/CD	<i>Carrier Sense Multiple Access with Collision Detection</i>
CTS	<i>Clear to Send</i>
DAC	<i>Digital Analog Converter</i>
DCF	<i>Distributed Coordination Factor</i>
DDC	<i>Digital Down Converter</i>
DIFS	<i>DCF Interframe Space</i>
DS	<i>Distribution System</i>



DSP	<i>Digital Signal Processor</i>
DSSS	<i>Direct Sequence Spread Spectrum</i>
DUC	<i>Digital Up Converter</i>
EIFS	<i>Extended Interframe Space</i>
ESS	<i>Extended Service Set</i>
FCS	<i>Frame Check Sequence</i>
FER	<i>Frame Error Rate</i>
FHSS	<i>Frequency Hopping Spread Spectrum</i>
FPGA	<i>Field Programmable Gate Array</i>
GSM	<i>Global System for Mobile Communications</i>
HCF	<i>Hybrid Coordination Function</i>
IBSS	<i>Independent Basic Service Set</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IFFT	<i>Inverse Fast Fourier Transform</i>
IFS	<i>Interframe Space</i>
IR	<i>Infrared Light</i>
LabVIEW	<i>Laboratory Virtual Instrument Engineering Workbench</i>
LAN	<i>Local Area Network</i>
LPF	<i>Low Pass Filter</i>
LTE	<i>Long Term Evolution</i>
MAC	<i>Medium Access Control</i>
MIMO	<i>Multiple Inputs Multiple Outputs</i>
NACK	<i>Negative Acknowledgment</i>



NAV	<i>Network Allocation Vector</i>
OFDM	<i>Orthogonal Frequency Division Multiplexing</i>
PCF	<i>Point Coordination Function</i>
PER	<i>Packet Error Rate</i>
PHY	<i>Physical</i>
PIFS	<i>PCF Interframe Space</i>
PLCP	<i>PHY Layer Convergence Procedure</i>
PPP	<i>Post Processor</i>
PSDU	<i>PLCP Service Data Unit</i>
QoS	<i>Quality of Service</i>
QPSK	<i>Quadrature Phase Shift Keying</i>
RAM	<i>Random Access Memory</i>
RTS	<i>Request to Send</i>
SDR	<i>Software Defined Radio</i>
SIFS	<i>Short Interframe Space</i>
STA	<i>Stations</i>
USRP	<i>Universal Software Radio Peripheral</i>
Wi-Fi	<i>Wireless Fidelity</i>
WLAN	<i>Wireless Local Area Network</i>



Bibliografía

- [1] IEEE Standard 802.11-2013 for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.
- [4] A. Di Stefano, G. Fiscelli, C.G. Giaconia, “An FPGA-Based Software Defined Radio Platform for the 2.4 GHz ISM Band”, Research in Microelectronics and Electronics, IEEE, 2006.
- [2] Takashi Shono, Yushi Shirato, Hiroyuki Shiba, Kazuhiro Uehara, Katsuhiko Araki, Masahiro Umehira, “IEEE 802.11 Wireless LAN Implemented on Software Defined Radio With Hybrid Programmable Architecture”, IEEE Transactions On Wireless Communications, Vol. 4, No. 5, SEPTEMBER 2005.
- [3] Lin LEI; Chen SONG, Taige Zhang, “Performance evaluation for OFDM Link Based on LabView and USRP”, 4TH International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), IEEE, 2014.
- [4] P. Fuxjager, A. Constantini, D. Valerio, P. Castiglione, G. Zacheo, T. Zemen, F. Ricciato, “IEEE 802.11p Transmission Using GNURadio”, 6th Karlsruhe Workshop on Software Radios (WSR), 2010.
- [5] B. Bloessl, M. Segata, C. Sommer and F. Dressler, “An IEEE 802.11a/g/p OFDM Receiver for GNU Radio”, ACM SIGCOMM 2013, 2nd ACM SIGCOMM Workshop of Software Radio Implementation Forum, 2013.
- [6] J. R. Gutiérrez-Agullo, B. Coll-Perales and J. Gozálvez, “An IEEE 802.11 MAC Software Defined Radio Implementation for Experimental Wireless Communications and Networking Research”, Wireless Days, IEEE, 2010.
- [7] A. S. Tanenbaum, “Computer Networks”, Pearson, Fifth Edition, 2011.



- [8] Chia Horng-Liu, "On the Design of OFDM Signal Detection Algorithms for Hardware Implementation", Globecom 2003, IEEE, 2003.
- [9] R. Prasad, "OFDM for Wireless Communications Systems", Artech House, 2004.
- [10] "LabVIEW Fundamentals", National Instruments, 2005.
- [11] B.Sklar, "Digital Communications Fundamentals and Applications", Prentice Hall, Second Edition, 2001.
- [12] Y. Li, G. Stuber, "Orthogonal Frequency Division Multiplexing for Wireless Communications", Springer, 2006.
- [13] "USRP N200/N210 Networked Series Datasheet", Ettus Research, 2012.
- [14] <http://www.ni.com/labview/>
- [15] http://files.ettus.com/manual/page_usrp2.html
- [16] http://files.ettus.com/manual/page_images.html