



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

Implementación del protocolo 802.11p basado en SDR para los sistemas ITS

T E S I S
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
YILENA ABRAHANTES HURTADO

TUTOR PRINCIPAL
VICTOR RANGEL LICEA, FACULTAD DE INGENIERÍA

MÉXICO, D. F. NOVIEMBRE 2017

JURADO ASIGNADO:

Presidente: **Dr. Matías Maruri José María**
Secretario: **Dr. Gómez Castellanos Javier**
Vocal: **Dr. Rangel Licea Víctor**
1^{er.} Suplente: **Dra. Moumtadi Fátima**
2^{d o.} Suplente: **Dr. Gutiérrez Castrejón Ramón**

Lugar o lugares donde se realizó la tesis:

México D.F., Facultad de Ingeniería, UNAM.

TUTOR DE TESIS:

Dr. Victor Rangel Licea

FIRMA



AGRADECIMIENTOS

A mis padres...

...por ser mi apoyo y mi mayor motivación, gracias por formarme como la persona que soy, gracias por ese amor incondicional que cruza fronteras, gracias por esa confianza que han depositado en mí, gracias por inspirarme a seguir mis sueños, gracias por ser exactamente como son...este logro es por y para ustedes, los amo.

A mi pareja...

...por atreverse a cruzar conmigo esta travesía, por la gran paciencia que eso conlleva, por esas noches de desvelo y preocupación que nos acompañaron estos dos años, por haber crecido juntos y permitirnos superar todo tipo de retos. A ti, Rubén...por todo ese amor que me has brindado: muchas gracias.

A mi familia...

...que aún lejos, no ha dejado de brindarme su apoyo.

A mis amigos...

...por ser mi soporte emocional, mi agradecimiento sobre todo aquellos con los que comparto la experiencia de vivir en un país extranjero...en especial, a mi amiga de muchos años con la que sé podré contar siempre: Yenny.

A la Universidad Nacional Autónoma de México...

...por darme una de las oportunidades más grandes de mi vida, por esa gran formación que ha permitido complementar mi desarrollo profesional, por permitirme conocer excelentes personas que llevaré siempre en una parte de mi corazón... Hoy siento que le voy a los Pumas y que "Soy hecha en CU", en fin... gracias por permitirme sentir ese orgullo de ser UNAM.

A mi tutor...

...por sus enseñanzas, por la confianza depositada en mí y sobre todo, por permitirme ser parte de su grupo de trabajo.

Al Consejo Nacional de Ciencia y Tecnología...

...por el apoyo económico otorgado para poder realizar esta Maestría.

A la DGAPA-UNAM...

...por el apoyo otorgado al proyecto de investigación PAPIIT IN116316 "Diseño y evaluación de técnicas de calendarización aplicadas a los Sistemas de Transporte Inteligente".



RESUMEN

La tecnología SDR, aunque lleva algunos años en desarrollo, aún se encuentra en pleno proceso de estudio e investigación. Unos de los principales desarrollos en el área ha sido la implementación de la capa física de distintas tecnologías inalámbricas que permitan medir el desempeño de las plataformas SDR y obtener sistemas de comunicaciones de una manera económica y eficiente. Por otra parte, las investigaciones y el estándar para VANETs, IEEE 802.11p, no incluye especificaciones para una versión MIMO, pero debido a que el canal de radio en VANETs es muy variable, se supone que los sistemas MIMO pueden ser muy prometedores para las redes vehiculares.

Debido a lo anterior y la flexibilidad de SDR, en la presente tesis se implementa una versión MIMO de la capa física del estándar 802.11p y se compara con el protocolo 802.11p tradicional, cuya implementación se encuentra previamente disponible como un paquete modular de código abierto.

En esta investigación se lleva a cabo la creación del prototipo de la capa física del estándar 802.11p utilizando las técnicas OFDM y MIMO con codificación de Alamouti en un esquema 2x1, por ende, se diseñan y desarrollan los bloques necesarios para que la señal sea procesada en el entorno multiantena propuesto.

El funcionamiento de dicho sistema es verificado mediante la realización de pruebas de laboratorio con señales RF en tiempo real en dos escenarios interiores distintos utilizando las modulaciones BPSK 1/2, QPSK 1/2, 16 QAM 1/2 y 64 QAM 2/3, Para determinar los resultados en cada uno de los casos (sistema MIMO y sistema SISO), se enviaron 1000 paquetes de un total de 100 bytes (72 de payload), cada 200 milisegundos. Cada una de las pruebas fueron realizadas cinco veces y, posteriormente, se determinó el promedio de los resultados obtenidos en las mismas tales como: la razón de paquetes recibidos, la razón de error de bits en cada una de las corridas y el SNR recibido en cada uno de los paquetes.



ABSTRACT

SDR technology has been in development for some years, and it is still in the process of study and research. One of the main developments in this area has been the implementation of the physical layer of different wireless technologies that allows it to measure the performance of SDR platforms and obtain communications systems in an economical and efficient way. On the other hand, the research and standard for VANETs, IEEE 802.11p, does not include specifications for a MIMO version, but because the radio channel in VANETs is highly variable, it is assumed that MIMO systems can be very promising for vehicular networks.

Due to the above and the flexibility of SDR, in the present thesis, a MIMO version of the physical layer of the 802.11p standard is implemented and compared with the traditional 802.11p protocol, whose implementation is previously available as an open source modular package.

In this research, the creation of the prototype of the physical layer of the 802.11p standard is performed using the OFDM and MIMO techniques with Alamouti coding in a 2x1 scheme, therefore, the blocks necessary for the signal to be processed are designed and developed in the proposed multi-antenna environment.

The operation of this system is verified by conducting laboratory tests with RF signals in real time in two different indoor scenarios using the BPSK 1/2, QPSK 1/2, 16 QAM 1/2 and 64 QAM 2/3 modulations. To determine the results in each case (MIMO system and SISO system), 1000 packets were sent out of a total of 100 bytes (72 payload) every 200 milliseconds. Each of the tests were performed five times, and afterwards, the average of the results obtained in them were determined, such as: the received packet ratio, the bit error ratio in each of the runs, and the SNR received in each of the packets.



CONTENIDO

RESUMEN.....	III
CONTENIDO	V
LISTA DE FIGURAS	VIII
LISTA DE TABLAS	X
ACRÓNIMOS.....	XI
CAPÍTULO 1. INTRODUCCIÓN	1
1.1 Antecedentes.....	1
1.2 Definición del problema	2
1.3 Objetivos.....	3
1.4 Contribuciones y relevancia	4
1.5 Estructura de la tesis	4
CAPÍTULO 2. ESTADO DEL ARTE	6
2.1 Estándar 802.11	6
2.2 Trabajos relacionados.....	7
CAPÍTULO 3. ESTÁNDAR IEEE 802.11p Y TECNOLOGÍA MIMO	12
3.1 Componentes de la arquitectura 802.11 [16]	12
3.1.1 Interfaces de servicios lógicas	14
3.1.2 Tipos de movilidad	15
3.2 Formatos del frame MAC [16].....	16
3.3 Estándar WAVE.....	18
3.4 Especificación de la capa PHY OFDM para la banda de 5GHz.	19
3.4.1 Funciones de la capa física (PHY) OFDM.....	20
3.4.2 Subcapa PLCP OFDM.....	22
3.4.3 Proceso de codificación del PPDU	23



3.5	MIMO (<i>Multiple Input Multiple Output</i>).....	25
3.5.1	Codificación espacio-temporal.....	26
3.5.2	Esquema Alamouti.....	27
3.6	Estimación del canal.....	28
3.6.1	Estimación del canal asistido por pilotos.....	29
CAPÍTULO 4. <i>Software Defined Radio</i> (SDR).....		33
4.1	Introducción a la plataforma SDR.....	33
4.2	<i>Universal Software Radio Peripheral</i> (USRP).....	35
4.2.1	USRP N210.....	35
4.2.2	USRP X310.....	37
4.3	Tarjeta hija.....	39
4.3.1	SBX.....	39
4.3.2	CBX-120.....	39
CAPÍTULO 5. GNU RADIO.....		41
5.1	Introducción.....	41
5.2	<i>GNU Radio Companion</i> (GRC).....	43
5.2.1	Interfaz gráfica de usuario QT.....	45
5.2.2	Modulación Digital.....	45
5.3	OFDM en GNU Radio.....	46
5.3.1	Transmisión.....	46
5.3.2	Recepción.....	47
5.3.3	Transmisión de paquetes.....	47
5.4	Interfaz <i>USRP Hardware Driver</i> (UHD).....	47
CAPÍTULO 6. DESARROLLO.....		49
6.1	Sistema SISO 802.11p.....	49
6.1.1	Transmisor.....	49
6.1.2	Receptor.....	51
6.2	Sistema MIMO propuesto.....	54
6.3	Implementación en software.....	55
6.3.1	Transmisor.....	55
6.3.2	Receptor.....	58
6.4	Implementación en Hardware.....	59
CAPÍTULO 7. RESULTADOS.....		61
7.1	Recolección de resultados.....	61
7.2	Especificaciones del sistema.....	63



7.2.1	Escenario 1.....	66
7.2.2	Escenario 2.....	70
CONCLUSIONES		76
TRABAJO FUTURO		78
REFERENCIAS		79
ANEXOS.....		82
Anexo A. Hoja de datos USRP N210.....		82
Anexo B. Hoja de datos USRP X310.....		84
Anexo C. Procesamiento de datos en Matlab tomados desde un libro Excel.....		86



LISTA DE FIGURAS

Fig 3.1 Conjunto de Servicio Básico (BSS) establecido en el estándar IEEE 802.11 [16].	13
Fig 3.2 Conjunto de servicios extendidos (ESS) [16].	14
Fig 3.3 Arquitectura completa IEEE 802.11[16].	15
Fig 3.4 Formato de una trama MAC [17].	16
Fig 3.5 Estructura del campo Frame Control [17].	17
Fig 3.6 Estructura del estándar 802.11[19].	20
Fig 3.7 Secuencias de entrenamiento que conforman el preámbulo PLCP	21
Fig 3.8 Formato de trama PPDU [16].	22
Fig 3.9 Esquema de diversidad con dos antenas en transmisión y una en recepción [22].	27
Fig 3.10 Sistema con codificación espacio-temporal OFDM con dos antenas transmisoras	29
Fig 3.11 Arreglo de pilotos tipo bloque [27].	30
Fig 3.12 Arreglo de pilotos tipo combinado [27].	30
Fig 4.1 Diagrama en bloques de SDR [35].	34
Fig 4.2 Arquitectura de la USRP N210.	36
Fig 4.3 Arquitectura de la USRP X310 [38].	38
Fig 4.4 Tarjeta hija SBX 400-4400 MHz Rx/Tx (40 MHz) [39].	39
Fig 4.5 CBX 1200-6000 MHz Rx/Tx (120 MHz), solo serie X [40].	40
Fig 5.1 Arquitectura del Software de GNU Radio.	42
Fig 5.2 Código de colores de datos en GNU Radio.	43
Fig 5.3 Estructura de la pantalla de inicio del GRC [42].	44
Fig 6.1 Transmisor implementado en GNU Radio del sistema SISO.	50
Fig 6.2 Esquema del receptor en GNU Radio del sistema SISO.	51
Fig 6.3 Esquema del sistema MIMO propuesto.	55
Fig 6.4 Implementación en GNU Radio del transmisor propuesto.	56
Fig 6.5 Patrón ortogonal de pilotos.	57
Fig 6.6 Configuración MIMO del bloque UHD: USRP Sink.	57



Fig 6.7 Implementación en GNU Radio del receptor propuesto.....	59
Fig 6.8 Diagrama general de la implementación MISO.	60
Fig 6.9 Cable de expansión MIMO utilizado en la implementación.....	60
Fig 7.1 Bloque de encapsulación RFtap.....	62
Fig 7.2 Captura de un paquete 802.11 con encapsulación RFtap.	62
Fig 7.3 Antena VERT2450.	64
Fig 7.4 Ocupación de canales en la banda 2.4 GHz en el entorno de prueba.	65
Fig 7.5 Primer escenario para la prueba del sistema MIMO.	67
Fig 7.6 Gráfica de BER vs SNR para modulación BPSK 1/2.	68
Fig 7.7 Gráfica de BER vs SNR para modulación QPSK 1/2.	68
Fig 7.8 Gráfica de BER vs SNR para modulación 16QAM 1/2.	69
Fig 7.9 Gráfica de BER vs SNR para modulación 64 QAM 2/3.	70
Fig 7.10 Implementación en hardware del sistema SISO (izquierda) y sistema MISO (derecha).	71
Fig 7.11 Gráfica de BER vs distancia para modulación BPSK 1/2.	72
Fig 7.12 Gráfica de BER vs distancia para modulación QPSK 1/2.	73
Fig 7.13 Gráfica de BER vs distancia para modulación 16 QAM 1/2.	74
Fig 7.14 Gráfica de BER vs distancia para modulación 64 QAM 2/3.	75



LISTA DE TABLAS

Tabla 3.1 Parámetros dependientes de la tasa de transmisión [16]	25
Tabla 3.2 Secuencia de codificación y transmisión para un esquema de diversidad de dos antenas transmisoras.	28
Tabla 4.1 Especificaciones del USRP N210 [37].	36
Tabla 4.2 Comparación entre las FPGA del USRP N210 y USRP X310.	38
Tabla 5.1 Bloques de visualización QT en GNU Radio [43].	45
Tabla 7.1 Parámetros del sistema de transmisión.	63
Tabla 7.2 Parámetros de sistema de recepción.	64
Tabla 7.3 Promedio de paquetes decodificados correctamente en relación al promedio del SNR recibido para BPSK $\frac{1}{2}$	67
Tabla 7.4 Promedio de paquetes decodificados correctamente en relación al promedio del SNR recibido para QPSK $\frac{1}{2}$	68
Tabla 7.5 Promedio de paquetes decodificados correctamente en relación al promedio del SNR recibido para 16QAM $\frac{1}{2}$	69
Tabla 7.6 Promedio de paquetes decodificados correctamente en relación al promedio del SNR recibido para 64 QAM $\frac{2}{3}$	69
Tabla 7.7 Promedio de paquetes recibidos en relación a la distancia para BPSK $\frac{1}{2}$	71
Tabla 7.8 Promedio de paquetes recibidos en relación a la distancia para QPSK $\frac{1}{2}$	72
Tabla 7.9 Promedio de paquetes recibidos en relación a la distancia para 16 QAM $\frac{1}{2}$	73
Tabla 7.10 Promedio de paquetes recibidos en relación a la distancia para 64 QAM $\frac{2}{3}$	74



ACRÓNIMOS

ACK	Acknowledgement
ADC	Analog to Digital Converter
AGC	Automatic Gain Control
AP	Access Point
API	Simplified Wrapper and Interface Generator
BER	Bit Error Rate
BSA	Basic Service Area
BSS	Basic Service Set
BSSID	Basic Service Set Identification
BPSK	Binary Phase Shift Keying
CCA	Clear channel assessment
CKK	Complementary Code Keying
CPU	Core Processing Unit
CRC	Cyclic Redundancy Check
CSI	Channel State Information
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
CTS	Clear to Send
DA	Destination Address
DAC	Digital to Analog Converter
DC	Direct Current
DCF	Distributed Coordination Function
DFS	Dynamic Frequency Selection
DFT	Discrete Fourier Transform



DLT	Data Link Type
DS	Distribution System
DSE	Dynamic Station Enablement
DSM	Distribution System Medium
DSP	Digital Signal Processing
DSRC	Dedicated Short Range Communications
DSS	Distribution System Service
DSSS	Direct Sequence Spread Spectrum
ESS	Extended Service Set
ETSI	European Telecommunications Standard Institute
FCC	Federal Communications Commissions
FCS	Frame Check Sequence
FHSS	Frequency Hopping Spread Spectrum
FM	Frecuencia Modulada
FPGAs	Field Programmable Gate Arrays
GRC	GNU Radio Companion
GPIO	General Purpose Input/Output
GPS	Global Positioning System
GPSDO	Global Positioning System - disciplined oscillators
IBSS	Independent Basic Service Set
IEEE	Institute of Electrical and Electronic Engineers
IFFT	Inverse fast Fourier transform
ISM	Industrial, Scientific and Medical
ITS	Intelligent Transport System
JTAG	Joint Test Action Group
LAN	Local Area Network
LLC	Link Layer Control
LOS	Line of Sight
LS	Least Square
LTE	Long Term Evolution
MAC	Media Access Control
MANET	Mobile Ad-Hoc Network



MIMO	Multiple-input Multiple-output
MSDU	MAC Service Data Unit
OBU	On-board Unit
OFDM	Orthogonal Frequency Division Multiplexing
PC	Personal Computer
PCIe	Peripheral Component Interconnect Express
PPDU	PLCP Protocol Data Unit
PER	Packet Error Rate
PLCP	Physical layer convergence procedure
PMD	Physical Medium Dependant
PSDU	PLCP service data unit
PSK	Phase Shift Keying
QAM	Quadrature amplitude modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RA	Receiver Address
RF	Radio Frequency
RSU	Roadside Unit
RTS	Request to Send
SA	Source Address
SDR	Software Defined Radio
SISO	Single-in Single-out
SNR	Universal Software Radio Peripheral
SoC	System-on-chip
SS	Station Service
SSID	Service Set Identification
SSK	Space Shift Keying
STA	Station
SWIG	Simplified Wrapper and Interface Generator
TA	Transmitter Address
TC	Traffic Category
TPC	Transmit Power Control



TS	Traffic Stream
UHD	USRP Hardware Driver
USRP	Universal Software Radio Peripheral
VANET	Vehicular Ad-Hoc Network
V-BLAST	Vertical BLAST
VOLK	Vector Optimized Library of Kernels
WAVE	Wireless Access for Vehicular Environments
WBSS	Wave Basic Service Set
WEP	Wireless Equivalent Privacy
Wi-Fi	Wireless Fidelity
WiGig	Wireless Gigabit Alliance
WLAN	Wireless Local Area Network



CAPÍTULO 1. INTRODUCCIÓN

1.1 Antecedentes

El área de redes de transporte inteligente o “ITS” por sus siglas en inglés, es una de las áreas de investigación de las tecnologías de la información y la comunicación que más relevancia ha tomado en los últimos años. El crecimiento de la industria automovilística y la demanda de vehículos a nivel mundial, hacen que éste sistema de transporte se convierta en un amplio campo para la implementación de tecnologías de telecomunicaciones que permitan el intercambio de información entre vehículos con la finalidad de aumentar la seguridad, la eficiencia en el manejo del tráfico, el control de emisión de gases de efecto invernadero, entre otras. Estas aplicaciones serán viables una vez que las redes vehiculares sean implementadas, ya sea que se desplieguen entre vehículos o a través de una combinación de vehículos con una infraestructura fija de telecomunicaciones. El potencial que exponen este tipo de comunicaciones, convierten a los sistemas de transporte inteligente en un área de desarrollo e investigación de suma importancia que involucra tanto a gobiernos a través de sus agencias de tránsito, centros universitarios destinados a la investigación y operadores de telecomunicaciones.

Uno de los componentes más importantes de las redes de transporte inteligentes, son las redes vehiculares *Ad-Hoc* o “VANET”, las cuales permiten la comunicación y el intercambio de información entre vehículos. Las VANET son un caso particular de las llamadas redes móviles *Ad-Hoc* o “MANET” que se pueden describir de manera básica como redes cuyo comportamiento es dinámico y relativamente espontáneo debido a la movilidad de sus nodos. Las VANET conforman la parte central de los sistemas ITS y toman ventaja de elementos actualmente implementados en los vehículos modernos como sistemas de posicionamiento global o “GPS” así como de una variedad de sensores y actuadores, los cuales pueden ayudar al conductor a tomar mejores decisiones en determinadas condiciones de manejo. Debido a las ya mencionadas aplicaciones y al alcance de éstos sistemas de comunicación, se ha desarrollado y estandarizado por parte de la IEEE un estándar exclusivo para la implementación de redes vehiculares denominado IEEE 802.11p, *Wireless Access for Vehicular Environments* (WAVE por sus siglas en inglés) o *Dedicated Short Range Communications* (DSRC, por sus siglas en inglés), y considera aspectos específicos a ser tomados en cuenta a nivel de toda la pila de protocolos del estándar. Sin embargo, el campo de investigación en éste tipo de redes se encuentra en una fase de desarrollo prematuro en lo que se refiere a pruebas de campo, así como en la



implementación de comunicaciones entre vehículos e infraestructura fija de telecomunicaciones, este último aspecto podría llegar a incrementar aún más las posibles aplicaciones de éstas redes.

1.2 Definición del problema

En torno a las comunicaciones inalámbricas se han realizado gran número de investigaciones que han conllevado a la estandarización e implementación de protocolos cuyas tecnologías se han expandido a nivel mundial, formando parte de la cotidianidad de un gran número de personas. Tales tecnologías, se han desarrollado tradicionalmente en sistemas de propósitos específicos basados en hardware, siendo prácticamente imposible su modificación. Sin embargo, con el desarrollo de FPGAs (*Field Programmable Gate Arrays*) y *system-on-chip* (SoC) programables para realizar funciones digitales se pueden realizar prototipos de tecnologías inalámbricas que van desde FM hasta Wi-Fi y LTE, cuyas configuraciones de modulación, codificación, ganancia, ancho de banda y demás, se realizan a través de configuraciones en software.

Ello dio lugar al concepto definido por Joseph Mitola [1] denominado Radios Definidos por Software (SDR, por sus siglas en inglés), permitiendo la implementación de las características y funciones de la capa física de varias tecnologías inalámbricas en una plataforma programable y reconfigurable, ofreciendo ventajas como múltiples modos de operación, reconfiguración, actualizaciones *Over-The-Air* y menor coste de desarrollo.

La tecnología SDR, aunque lleva algunos años en desarrollo, aún se encuentra en pleno proceso de estudio e investigación. Unos de los principales desarrollos en el área ha sido la implementación de la capa física de distintas tecnologías inalámbricas que permitan medir el desempeño de las plataformas SDR y obtener sistemas de comunicaciones de una manera económica y eficiente.

Por ende, debido a la flexibilidad de SDR, se decide implementar la capa física de un sistema basado en el estándar 802.11p, el cual básicamente difiere muy poco de las versiones 802.11a/g. Dicha implementación será realizada en GNU Radio, ofreciendo la posibilidad de implementar y evaluar distintos algoritmos que ayuden a obtener un sistema eficiente tanto en transmisión como en recepción y una comunicación con un bajo índice de BER (*Bit Error Rate*). Ello será implementado utilizando como radios los equipos Ettus USRP X310 y N210 en un sistema de comunicaciones MIMO (*multiple-input multiple-output*) introduciendo un bloque de codificación espacio-temporal, con lo cual se pretende lograr mejores resultados que los prototipos anteriores.

Tal desarrollo tendrá como base el sistema SISO 802.11p ya implementado en [10], cuyo código se encuentra disponible como un paquete modular de bloques de construcción completamente de código abierto.

Las investigaciones y el estándar para VANETs, IEEE 802.11p, no incluye especificaciones para una versión MIMO, pero debido a que el canal de radio en VANETs es muy variable, se supone que los sistemas MIMO pueden ser muy prometedores para las redes vehiculares. Debido a ello, se



implementa una versión MIMO del estándar 802.11p y se compara con el protocolo 802.11p tradicional.

La idea de implementar el estándar que se ocupa en el presente trabajo, se encuentra justificada debido a que los sistemas DSRC (*Dedicated Short Range Communication*) están considerados como la tecnología clave para sistemas de comunicación intervehicular a ser desarrollados en el futuro [2]. LA FCC (*Federal Communications Commissions*) en Estados Unidos y el ETSI (*European Telecommunications Standard Institute*) en Europa han asignado 75 MHz y 30 MHz respectivamente, en la banda de los 5.9 GHz para el uso de los ITS.

Los estándares para DSRC reúnen los requerimientos de las aplicaciones que se encargan de transferir información tanto entre los vehículos y dispositivos localizados a las orillas de las carreteras, como entre los mismos vehículos. Típicamente, éste tipo de comunicación ocurre entre vehículos en movimiento que entran a una zona de comunicación que cuenta con equipos de comunicaciones fijos localizados en las carreteras, o directamente entre vehículos en movimiento. Los sistemas DSRC proveen la base para una variedad de aplicaciones que incluyen seguridad de vehículos, peaje automatizado, navegación mejorada, gestión de tráfico y muchos otros [3].

1.3 Objetivos

El objetivo primordial de esta tesis se basa en la evaluación, implementación en equipos de radios definidos por software (SDR) y análisis del comportamiento dinámico del protocolo de comunicación WAVE-IEEE 802.11p, considerados en los Sistemas de Transporte Inteligente (ITS), en un entorno multiantena.

Tal propósito se pretende lograr utilizando la herramienta de desarrollo de software libre GNU Radio, la cual provee el procesamiento digital de las señales, así como bloques de procesamiento con funciones predefinidas que permitan implementar el sistema en plataformas rentables. Específicamente se utilizará la herramienta gráfica *GNU Radio Companion* para la creación de gráficos de flujos de señales y generar el código fuente de los gráficos de flujo.

Se llevará a cabo la creación del prototipo de la capa física del estándar 802.11p utilizando las técnicas OFDM y MIMO (*Multiple-input Multiple-output*) con codificación de Alamouti en un esquema 2x1, por ende, se deberán diseñar y desarrollar los bloques necesarios para que la señal sea procesada correctamente en el entorno multiantena propuesto.

Se verificará el funcionamiento del sistema mediante la realización de pruebas de laboratorio con señales RF en tiempo real, se validará el correcto procesamiento de los paquetes en la implementación MIMO, y finalmente se compararán los resultados con la implementación SISO (*Single-input Single-output*) para medir su desempeño, la viabilidad de implementarlo en tiempo real y así lograr una caracterización del sistema.



1.4 Contribuciones y relevancia

Las plataformas USRPs han sido ampliamente utilizadas en entornos de pruebas para realizar evaluaciones experimentales de diferentes protocolos de comunicaciones y de sistemas MIMO, debido a que es una alternativa más económica y flexible que la encontrada en los entornos de pruebas tradicionales. Por ende, realizar la caracterización de un sistema MIMO utilizando dichas plataformas, en conjunto con la herramienta de desarrollo de software libre GNU Radio, permiten evaluar el desempeño del sistema de una manera efectiva y menos costosa que los equipos analizadores de redes utilizados normalmente. La presente investigación, por tanto, tiene un elevado impacto práctico en los entornos investigativos de los sistemas inalámbricos.

Con la realización del estado del arte de la misma se pretende dar un panorama general de la plataforma SDR, su desarrollo y el estado de la tecnología a la fecha. Además, se pretende contribuir a la obtención de un algoritmo eficiente para lograr la transmisión y recepción de señales en sistemas MIMO utilizando SDR.

En esta tesis se desarrolla exitosamente la capa física del estándar IEEE 802.11p en combinación con un sistema MIMO 2x1 utilizando codificación Alamouti en GNU Radio como software y dos USRP N210 para la transmisión y un USRP X310 para la recepción, en la parte de hardware. Hasta donde se conoce, se implementará el primer sistema basado en la capa PHY del estándar 802.11p utilizando la técnica MIMO y abarcando hasta una modulación de 64QAM, por lo que se presenta un primer acercamiento utilizando la tecnología SDR que sirva de base para realizar estudios relacionados con esta técnica en entornos vehiculares.

Al final del proyecto, se tendrá un estudio del desempeño de un sistema de comunicaciones MIMO comparado con un sistema SISO bajo el estándar de la capa física del estándar 802.11p y se analizará la viabilidad para implementarlo en ambientes reales mediante las mediciones de Razón de Paquetes Recibidos.

1.5 Estructura de la tesis

Se propone dividir el trabajo en 3 etapas fundamentales, con objetivo de dar continuidad a la investigación realizada en [4, 5], en un primer momento se pretende llevar a cabo la implementación de la capa física del estándar 802.11p tanto en transmisión como en recepción utilizando la plataformas Ettus USRP N210 y X310 y la herramienta de software libre GNU Radio. Tal implementación se encuentra disponible en [10] como un paquete modular de bloques de construcción completamente de código abierto, y sirvió como base en la presente investigación.

En un segundo momento se extenderá el sistema de comunicaciones a un esquema MIMO con dos antenas en transmisión y una antena en recepción, utilizando el esquema de codificación espacio-temporal Alamouti. Se realizará la programación de los bloques de codificación y decodificación Alamouti en transmisión y recepción respectivamente, implementando, por ende, un sistema MIMO



2x1 utilizando dos USRP N210 sincronizados en transmisión y un USRP X310 para la recepción.

Finalmente, se verificará el funcionamiento del sistema MIMO para medir parámetros como BER vs Distancia y BER vs SNR en comparación con el sistema SISO.

Para realizar lo expuesto anteriormente, se decide fragmentar el presente trabajo en 7 capítulos. En el capítulo 2 se brinda una introducción al estándar 802.11 y su estado actual en la implementación con SDR, además de abordar los estudios relacionados con relación a la tecnología MIMO y su aplicación en entornos de comunicación vehicular.

En el capítulo 3 se ofrece una descripción del estándar IEEE 802.11p, enfatizando en los detalles de la capa física (PHY) ya que constituye un tema esencial en la presente investigación. Además, se ofrece un panorama general de la tecnología MIMO, los diferentes tipos de codificaciones que puede comprender y el funcionamiento del esquema de Alamouti, ya que fue el implementado en este proyecto. Finalmente, se describe el algoritmo de estimación de canal utilizado, ya que dicho tema constituye un aspecto esencial en la utilización de sistemas MIMO.

En el capítulo 4 se ofrece una descripción detallada de la plataforma SDR, sus componentes principales y se brinda la descripción de los equipos USRP (*Universal Software Radio Peripheral*) utilizados: el USRP N210 y el USRP X310 de *Ettus Research*. Se describen sus arquitecturas, componentes y características.

El capítulo 5 contiene una introducción a la herramienta de software libre GNU Radio usado en este proyecto, se estudian sus características principales, se especifican las librerías más importantes utilizadas para implementar OFDM, así como la interfaz encargada de comunicarse con los dispositivos de hardware USRP.

En el capítulo 6 se presenta la descripción de la propuesta de este trabajo. Primeramente se describe el sistema SISO que sirvió como base de comparación e implementación para la presente investigación, los bloques que lo conforman tanto en transmisión como en recepción, así como sus funcionalidades. Posteriormente, se describe la implementación MIMO propuesta, cómo se llevó a cabo el proceso de codificación y decodificación con el esquema de Alamouti y las características y limitantes del sistema implementado con SDR.

Finalmente, el objetivo del capítulo 7 fue presentar los resultados obtenidos en dos escenarios en interiores, utilizando cuatro esquemas de modulación y codificación distintos. Para dichos escenarios se presentan las pruebas realizadas cuyo objetivo principal fue presentar el desempeño del sistema SISO tradicional en comparación con el sistema MIMO propuesto. En el análisis se tomaron en cuentas los resultados obtenidos de PER, BER en función del SNR (*Signal to Noise Ratio*) y de la distancia.



CAPÍTULO 2. ESTADO DEL ARTE

Conocer la evolución de las redes pertenecientes al estándar 802.11 es de gran importancia en el desarrollo del presente proyecto, ya que constituyen un pilar crucial en los sistemas de comunicaciones actuales, permitiendo una conexión desde oficinas, residencias o lugares públicos dedicados a este fin. Además, se debe realizar un análisis de las implementaciones realizadas en plataformas SDR, utilizando preferentemente GNU Radio, de manera que se enfoque directamente al objetivo de este proyecto.

El presente capítulo se divide en 2 partes, una primera comprende los principales aspectos de las redes pertenecientes al estándar 802.11 y sus distintas versiones; mientras que en la segunda se expondrán los trabajos más relacionados con la investigación actualmente llevada a cabo.

2.1 Estándar 802.11

El 1997 fue creado el estándar 802.11 por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, por sus siglas en inglés). Con el tiempo y los avances de las tecnologías y necesidades de la sociedad, el estándar ha evolucionado agregando diferentes versiones que permitan que el mismo se acomode a los escenarios cada vez más exigentes en los que se implementan las WLAN.

Tales exigencias vienen dadas por las ventajas que las WLAN ofrecen en comparación con las LANs tradicionales, entre las mismas se pueden mencionar características como movilidad, reducción de costos, fiabilidad, escalabilidad y menor tiempo de operación.

A continuación, se presentan algunas de las versiones del estándar 802.11 y sus principales características:

- IEEE 802.11: constituye la versión original del estándar y actualmente se encuentra obsoleta. Define 3 tecnologías de capa física: Infrarrojo y dos formas de modulación de espectro expandido: FHSS (*Frequency Hopping Spread Spectrum*) y DSSS (*Direct Sequence Spread Spectrum*). Se definen tasas de transferencia de 1 y 2 Mbps en la banda de 2.4 GHz utilizando un ancho de banda de 20 MHz.



- IEEE 802.11a: publicado dos años más tarde, alcanza una velocidad máxima de transferencia de 54 Mbps mediante la técnica OFDM (*Orthogonal Frequency Division Multiplexing*), ocupa un ancho de banda de 20 MHz en la banda de 5 GHz, existiendo por ende, menor interferencia que la banda de 2.4 GHz, pero con la deficiencia de alcanzar una menor cobertura. Presenta modulación adaptativa dependiendo de las características del canal utilizado.
- IEEE 802.11b: creado también en 1999, pero trabaja en la frecuencia de 2.5 GHz. Utiliza la técnica de DSSS con CCK (*Complementary Code Keying*), ancho de banda de 20 MHz y logra alcanzar una tasa de transferencia de hasta 11 Mbps.
- IEEE 802.11g: publicado en el 2003 utiliza la banda de 2.4GHz permitiendo de esta forma compatibilidad con el estándar anterior ('b'), ancho de banda de 20 MHz, ocupa ambas técnicas: DSSS y OFDM, modulación adaptativa y alcanza velocidades de hasta 54 Mbps.
- IEEE 802.11n: publicado en el 2009, opera tanto en la banda de 2.4 GHz como en la de 5 GHz. Además de ocupar los 20 MHz de ancho de banda tradicional, utiliza también ancho de banda de 40 MHz, permitiendo alcanzar mayores velocidades de transferencia que tiene un máximo teórico de 150 y 600 Mbps. Ello se logra gracias a la introducción de la técnica MIMO. Es compatible con las versiones anteriores 'a', 'b' y 'g'.
- IEEE 802.11p: creada para aplicaciones automovilísticas en las *denominadas Wireless Access in Vehicular Environments (WAVE)/Dedicated Short Range Communications (DSRC)*. Las dos principales diferencias introducidas en 802.11p son la frecuencia de operación y el ancho de banda, ya que utiliza la banda de 5.9 GHz en vez de 2.4 GHz (802.11b/g/n) y 5 GHz (802.11a/n), ello asegura que no habrá interferencia con las redes *Wi-Fi* tradicionales ampliamente desplegadas en gran cantidad de entornos. Además, se usa un ancho de banda de 10 MHz en vez de los 20 MHz utilizados en los estándares tradicionales de *Wi-Fi*.
- 802.11ac: publicada en el 2013, opera en la banda de los 5 GHz y logra alcanzar velocidades de hasta 450 Mbps y 1.3 Gbps. Puede llegar a alcanzar una modulación de 256-QAM, utiliza la técnica MIMO que se puede extender hasta un arreglo de 8x8, y anchos de bandas de 20 MHz, 40 MHz, 80 MHz y 160 MHz. Es compatible con dispositivos que operen bajo los estándares 802.11a/n.
- IEEE 802.11ad: conocido como *WiGig*. Utiliza las bandas de operación 2.4 GHz, 5 GHz y 60 GHz con velocidades máximas teóricas de 7 Gbps. Compatible con estándares 802.11 a/b/g/n/ac.

2.2 Trabajos relacionados

La primera implementación que fue capaz de generar y transmitir frames del estándar IEEE 802.11p basado en OFDM fue realizado en [6] en el año 2010. Dicho estándar difiere de las versiones 802.11a y 802.11g en dos aspectos principales: la banda de operación (corrida para la versión 802.11p a 5.9 GHz y la duplicación de la duración de los símbolos OFDM desde 4 μ s a 8 μ s, por tanto el codificador propuesto en [6] tenía la capacidad de generar tramas de los protocolos 802.11a/g solo cambiando dos parámetros en el *front-end*: el factor de interpolación y la frecuencia de portadora.



Dicha implementación fue llevada a cabo utilizando USRP2 y una tarjeta hija XCVR2450 que soporta un rango de frecuencias de 2.4 a 2.5 GHz y de 4.9 a 5.9 GHz. En el mismo, como primer paso se creó un codificador de referencia para las tramas OFDM en MATLAB, utilizando las instrucciones detalladas en el documento estandarizado IEEE 802.11-2007 [7]. Posteriormente se portó progresivamente la codificación de MATLAB al marco GNU Radio, tomando como objetivo reusar los bloques de procesamiento de señales disponibles en GNU Radio siempre que fuera posible.

Finalmente se realizaron las mediciones usando varios métodos: primero se utilizó una segunda computadora conectada a otro USRP2 para realizar la conversión inversa y registrar la señal transmitida. Segundo, utilizando opciones de tiempo de símbolos acorde con 802.11g y una frecuencia de portadora en la banda ISM 2.4 GHz, se generaron *frames* que fueron recibidos correctamente por un *chipset* convencional 802.11g. Finalmente se comparó el espectro enviado por el USRP2 y un equipo comercial de *Atheros*, observando que el USRP generaba picos en ± 6 MHz, los cuales no estaban presentes en el equipo *Atheros*. El resultado mostró que el transmisor resultante basado en GNU Radio fue capaz de generar tramas OFDM totalmente compatibles con los estándares IEEE 802.11a, 802.11g y 802.11p.

En el trabajo realizado en [8] se presentó una aplicación de Radio Definida por Software completamente programable de la capa MAC del estándar IEEE 802.11 MAC que puede ser utilizado, configurado y modificado totalmente, para desarrollar técnicas avanzadas de redes y de comunicaciones entre capas. Para dicha investigación, la capa PHY usada fue la desarrollada en [6] con algunas modificaciones ligeras en las funciones de sensado de la portadora y la capacidad de transmisión de paquetes de diferentes tamaños; incluyéndose los provenientes de la capa MAC que involucra los mensajes RTS, CTS, ACK y los *frames* de *Beacon* 802.11. La capa MAC implementada llevó a cabo el servicio de contención o la Función de Coordinación Distribuida (DCF, por sus siglas en inglés), la cual está basado en la técnica CSMA/CA. Los resultados confirmaron el correcto comportamiento del protocolo MAC implementado durante el proceso de envío de los paquetes de datos, aunque se experimentaron ciertos retardos debido principalmente a la capacidad de procesamiento de la CPU de la PC utilizada en la implementación. En cuanto a la recepción, la capa MAC fue desarrollada de igual forma, aunque el enlace inalámbrico no fue puesto a prueba debido a que no se tenía desarrollada la capa PHY en la recepción.

Luego en el año 2012 [9] se presentó una implementación eficiente de la capa física de un receptor 802.11p por una plataforma SDR flexible. La tecnología elegida fue la plataforma *OpenAirInterface ExpressMIMO* desarrollada por Eurecom y Télécom ParisTech. Esta plataforma permite una fácil actualización a futuros estándares, permite un uso efectivo del espectro, movilidad, capacidad de red incrementada, el mantenimiento de la reducción de costos y un rápido desarrollo de nuevos servicios. Dicho trabajo constituyó el primer prototipo de un receptor 802.11p desarrollado en la plataforma *ExpressMIMO*. Los resultados mostraron que podía ser logrado un flujo de control centralizado y procesamiento en tiempo real para BPSK, QPSK y 16-QAM, sugiriendo ciertas mejoras para poder lograr un resultado favorable en 64-QAM.



En el 2013 se presentó en [10] un receptor OFDM completo implementado en GNU Radio y adaptado para operar con la USRP N210 *Ettus*. Constituyó el primer prototipo hasta esa fecha implementado en SDR basado en un receptor OFDM que soportara anchos de bandas mayores a 20 MHz, como contraparte del trabajo realizado por Fyxjäger et al. en [6]. La aplicabilidad de tal receptor se extendió a los estándares 802.11 a/g/p, soportando anchos de bandas tanto de 20 MHz para aplicaciones de *WiFi* como de 10 MHz para sistemas DSRC 802.11p, con antenas localizadas a 6 metros una de otra. El receptor comprimió tanto la capa física como el proceso de decodificación completa incluyendo la MAC, y el código utilizado se puso disponible como un paquete modular de bloques de construcción completamente de código abierto para proporcionar un sistema de fácil acceso que pueda ser utilizado como una herramienta para experimentar con nuevos algoritmos de procesamiento de señales. Las modulaciones utilizadas fueron BPSK y QPSK con codificación $\frac{1}{2}$ y $\frac{3}{4}$ respectivamente, no se emplearon modulaciones de orden superior, ya que el receptor estaba limitado a PSK (*Phase Shift Keying*) debido a la falta de implementaciones de algoritmos sofisticados de estimación de canal. En general, la parte de recepción se realizó en dos etapas principales: la primera parte, la detección del *frame* y la segunda parte, la decodificación del mensaje, la cual está conformada por la corrección en frecuencia y fase, la alineación del símbolo OFDM, estimación del canal y la decodificación del campo de señal y del campo de datos.

Las mediciones de las pruebas se realizaron utilizando tarjetas *WiFi* 802.11 a/g y dispositivos 802.11p, los resultados emitidos para este último demostraron que las codificaciones son bien soportadas en SDR y que se logra una razón de entrega de paquetes de 1 para altas potencias de transmisión, aunque como se mencionó anteriormente no se soportaron esquemas de modulación y codificación de orden superior.

El trabajo anteriormente mencionado fue extendido en [11], mediante la realización de un transceptor OFDM para el protocolo 802.11p soportando los cuatros esquemas de transmisión BPSK, QPSK, 16-QAM y 64-QAM. Constituyó el primero de su tipo y se demostró según las medidas de desempeño que dicho sistema corría sin problemas sobre una laptop simple. El sistema se usó en la simulación de capas PHY y MAC mediante la conexión del transmisor y receptor en GNU Radio, y tales simulaciones correspondieron muy bien a un modelo de error aceptado ampliamente para redes IEEE 802.11p. El transceptor fue validado mediante una comparación con un radio Unex DCMA-86P2 de nivel comercial, resultando en el desempeño exactamente esperado.

Sin embargo, debido a latencias incurridos por la comunicación entre el PC y el USRP no fue posible llevar a cabo la detección de portadora lógicamente en software, ya que esto introduce un gran punto ciego entre el momento en el que se detecta el medio y cuando finalmente se accede. Al igual que con el transmisor, se experimenta la latencia en el receptor en la comunicación entre la PC y el USRP. Ello hizo que fuera imposible cumplir con las duras restricciones de tiempo de RTS/CTS y *frames* de acuse de recibo que se utilizan para las transmisiones *unicast*. Tampoco permitió el soporte del último esquema de modulación/codificación estandarizado: 64-QAM $\frac{3}{4}$.



Se referencian las investigaciones realizadas en la Universidad Nacional Autónoma de México en [5] y [4], a las que se pretende dar continuación. En las mismas se implementaron un transmisor y un receptor respectivamente que soportara la capa física del estándar IEEE 802.11 en *LabVIEW* para SDR, utilizando USRP N210, constituyendo el primer receptor en *LabVIEW* para OFDM trabajando con SDR que soporta esquemas de modulación BPSK, QPSK, 16-QAM y 64-QAM, contando además en la capa física con funciones de *Interleaver*, *scrambler* y codificación convolucional. Los resultados emitieron valores de BER en distintos entornos para conocer la viabilidad del sistema para su uso comercial. La transmisión y la recepción simultánea utilizando SDR fue exitosa para la norma IEEE 802.11g implementada, aunque debido a la limitante del propio equipamiento USRP N210 en la capacidad de la interfaz que conecta con la PC, se trabajó solo con 2 MHz de ancho de banda. Las mediciones del BER fueron realizadas en entornos con diferentes antenas y a diferentes distancias, lo que entregó una gran cantidad de datos, los cuales mostraron que tanto en interiores como en exteriores a una distancia máxima de 16 m se pudo obtener un BER aceptable, donde máximo se logró obtener 9.99×10^{-4} a 1 m de distancia. Debido a la reducción del ancho de banda, se obtuvo un máximo de velocidad de 3.6 Mb/s utilizando modulación 64-QAM.

Entre las investigaciones relacionadas con la implementación de la tecnología MIMO utilizando plataformas USRPs se encuentra la realizada en el [12] donde se propone mejorar el desempeño de sistemas ITS/DSRC utilizando técnicas MIMO y la tecnología SDR. La arquitectura implementada consistió en primer lugar de un transmisor cuyas señales fluían desde el software GNU Radio instalado en una PC hacia la tarjeta madre del USRP (FPGA Altera Cyclone EP1C12) y enviado a través de dos tarjetas RF XCVR2450, las cuales se encargaban de convertir las señales digitales a señales analógicas que a su vez fluían a las antenas transmisoras VERT2450 utilizadas para enviar las señales analógicas por la interfaz aire. En el receptor, el procedimiento se realizó de forma inversa al explicado en el transmisor. Las tarjetas hijas XCVR2450 trabajan en las bandas 2.4 GHz y 5.9 GHz, siendo ésta última la utilizada en la implementación, que consistió en configurar el transmisor y el receptor con codificación y decodificación MIMO en un esquema de 2x2 antenas para compararlo con un esquema SISO (*single-input single-output*). El experimento evaluó el desempeño del sistema mediante los parámetros PER, BER y transferencia de datos (*throughput*). Los autores variaron solamente las distancias entre el RSU (*roadside unit*) y el OBU (*on-board unit*), cuyos resultados emitieron que el PER del esquema MIMO 2x2 contra la distancia entre el RSU y el OBU es menor que el sistema SISO convencional. Por otra parte, el BER se incrementaba cuando la distancia entre el RSU y el OBU aumentaba.

Tales resultados confirmaron que, con el sistema propuesto, la técnica MIMO y la tecnología SDR mejoraron el desempeño de sistemas tradicionales, ya que era capaz de lograr altas velocidades de datos en la comunicación y mayor exactitud en la misma, permitiendo una mejora en la confiabilidad ofrecida a los usuarios.

Por otra parte, en la investigación realizada en [13] se aplicaron diversas técnicas MIMO para mejorar el desempeño o la razón de datos en comunicaciones vehiculares basadas en el estándar IEEE 802.11p. Para ello, se comparó el desempeño de tres esquemas MIMO: Alamouti, BLAST vertical (V-



BLAST) y modulación por desplazamiento de espacio (SSK, por sus siglas en inglés). Dicha investigación fue realizada considerando la influencia de diferentes velocidades de manejo con línea de vista (LOS, por sus siglas en inglés) y sin línea de vista (NLOS, por sus siglas en inglés). Los resultados obtenidos demostraron que el esquema V-BLAST podría incrementar significativamente la razón de datos en comparación con los sistemas tradicionales de una sola antena, mientras que el esquema SSK y Alamouti podrían mejorar la confiabilidad y la robustez, logrando este último, establecer la comunicación incluso bajo condiciones de recepción ínfimas.



CAPÍTULO 3. ESTÁNDAR IEEE 802.11p Y TECNOLOGÍA MIMO

El estándar 802.11 es una colección de las especificaciones de la capa física (PHY) y de la capa de control de acceso al medio (MAC) para la implementación de WLAN en las bandas de frecuencias 2.4, 3.6, 5 y 60 GHz. El estándar y sus anexos proveen los fundamentos de la tecnología *Wi-Fi*.

IEEE 802.11p es uno de los estándares más recientes para incorporar el acceso inalámbrico en ambientes vehiculares (WAVE, por sus siglas en inglés). Este anexa algunas mejoras a la última versión de 802.11 para soportar aplicaciones de Sistemas de Transporte Inteligente (ITS) [14] e incluye el intercambio de datos entre vehículos a alta velocidad y entre vehículos e infraestructura en la banda licenciada ITS.

El estándar 802.11p fue creado principalmente para ser utilizado en entornos donde las propiedades de la capa física cambien rápidamente y donde se requiera intercambio de comunicaciones de corta duración. Por ende, el estándar tiene como propósito proveer el mínimo conjunto de especificaciones requeridas para asegurar la interoperabilidad entre dispositivos inalámbricos que intenten comunicarse en entornos de comunicaciones potencialmente cambiantes de manera rápida y en situaciones donde las transacciones deben ser completadas en fracciones de tiempos muy cortos.

El estándar 802.11p define los parámetros de operación requeridos para implementar servicios de transferencia de datos de alta velocidad en la banda de los 5.9 GHz perteneciente a Servicios de Radio en Sistemas de Transporte Inteligente (ITS-RS, por sus siglas en inglés). Las comunicaciones generalmente toman lugar sobre una distancia menor que 1000 metros con línea de vista entre las unidades localizadas a las orillas de las carreteras y los vehículos en movimientos o fijos, o entre vehículos en movimiento [15].

3.1 Componentes de la arquitectura 802.11 [16]

La arquitectura IEEE 802.11 consiste en varios componentes que interactúan para proveer una WLAN que soporte movilidad a las estaciones (STAs, por sus siglas en inglés) transparentemente a las capas superiores.

El conjunto de servicios básicos (BSS, por sus siglas en inglés) es el bloque de construcción básico de una LAN IEEE 802.11. En la figura 3.1 se muestran dos BSSs, cada uno de los cuales tienen como miembros dos STAs.

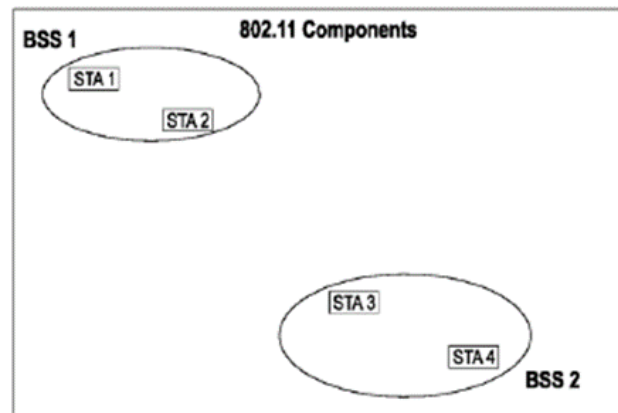


Fig 3.1 Conjunto de Servicio Básico (BSS) establecido en el estándar IEEE 802.11 [16].

Los óvalos usados para representar a los BSSs se pueden interpretar como áreas de coberturas en la cual los miembros STAs de los BSSs pueden permanecer en comunicación. Esta área se denomina Área Básica de Servicio (BSA, por sus siglas en inglés).

El BSS independiente (IBSS) es el tipo más básico de IEEE 802.11 LAN y puede contener un mínimo de dos STAs. Este modo de operación es posible cuando STAs IEEE 802.11 son capaces de comunicarse directamente y generalmente son referidas como redes ad-hoc.

Un BSS puede encontrarse aislado o puede estar conectado a un sistema de distribución (DS, por sus siglas en inglés) constituyendo un componente dentro la red que es construida con múltiples BSSs. El estándar separa lógicamente el medio inalámbrico del medio del sistema de distribución (DSM, por sus siglas en inglés), cada medio lógico es usado para propósitos diferentes, por componentes diferentes de la arquitectura.

Un punto de acceso (AP, por sus siglas en inglés) es cualquier entidad que tiene funcionalidades STA y permite el acceso al DS, a través del medio inalámbrico para los STAs asociados.

El DS y la infraestructura BSSs permiten crear una red inalámbrica de tamaño y complejidad arbitrarios, a lo cual el estándar se refiere como una red de conjunto de servicios expandidos (ESS, *Extended Service Set*). Un ESS es la unión de la infraestructura BSSs con un mismo SSID (*Service Set Identification*) conectado a través de un DS como se observa en la figura 3.2. Una red ESS contiene la misma estructura en la capa LLC que una red IBSS. Los STAs en un ESS se pueden comunicar y mover desde un BSS a otro (en el mismo ESS) transparentemente a la capa de control de enlace de datos (LLC, por sus siglas en inglés).

El BSSID (*Basic Service Set Identification*) en contraste al SSID, constituye el nombre de un BSS conocido a nivel de dirección MAC, y de la misma forma constituye un campo de 48 bits. Cada BSS debe tener un único BSSID compartido por todos los miembros. El filtrado BSSID es un mecanismo clave para restringir a nivel MAC todo los *frames* entrantes a solo los recibidos desde radios en el que todos los miembros están en el mismo BSS.

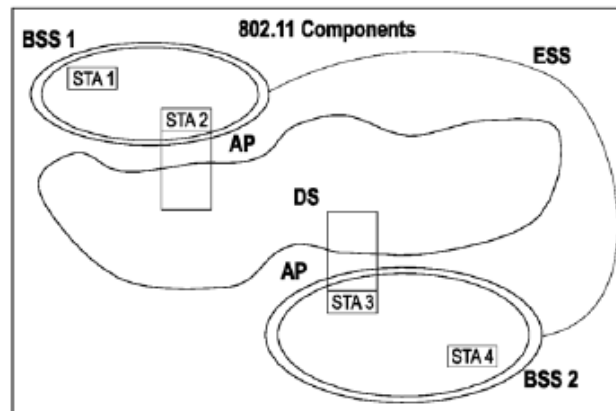


Fig 3.2 Conjunto de servicios extendidos (ESS) [16].

Para integrar la arquitectura IEEE 802.11 con una LAN alamburada tradicional, se introduce un componente denominado: portal. Un portal es el punto lógico al cual entra una unidad de datos de servicios MAC (MSDU) desde una LAN que no es IEEE-802.11 a un DS IEEE 802.11.

3.1.1 Interfaces de servicios lógicas

Un DS puede ser creado desde varias tecnologías incluyendo las LAN alambradas IEEE 802. El estándar 802.11 explícitamente no especifica los detalles de las implementaciones de los DS, pero especifica los denominados 'servicios'. Los servicios están asociados con diferentes componentes de la arquitectura. Existen dos categorías de servicios: los servicios de la estación (SS, por sus siglas en inglés) y los servicios del sistema de distribución (DSS, por sus siglas en inglés). Ambas categorías de servicios son usadas por la subcapa MAC IEEE 802.11.

El conjunto completo de los servicios de la arquitectura 802.11 se enumeran a continuación:

- a) Autenticación
- b) Asociación
- c) Desautenticación
- d) Desasociación
- e) Distribución
- f) Integración
- g) Confidencialidad de datos
- h) Reasociación

- i) Entrega de MSDU.
- j) Selección dinámica de frecuencias (DFS, por sus siglas en inglés)
- k) Control de potencia transmitida (TPC, por sus siglas en inglés)
- l) Sincronización de tiempo en capas superiores (solo facilidades QoS)
- m) Calendarización de tráfico QoS (solo facilidades QoS)
- n) Mediciones de radio
- o) Activación de estación dependiente (DSE, por sus siglas en inglés)

Este conjunto de servicios se divide en dos grupos: el SS y el DSS. El SS es parte de cada STA y el DSS es provisto por el DS.

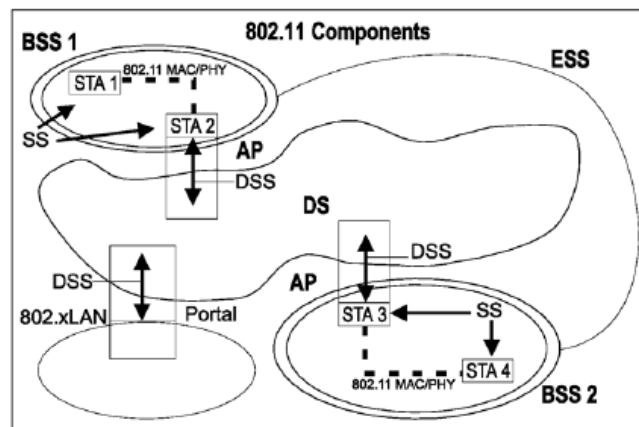


Fig 3.3 Arquitectura completa IEEE 802.11[16].

3.1.2 Tipos de movilidad

Los tres tipos de transición de mayor significado en esta norma que describen la movilidad de STAs dentro de una red son los siguientes [16]:

- a) Sin transición: se identifican dos subclases que son usualmente indistinguibles:
 - 1. Movimiento estático - sin movimiento.
 - 2. Movimiento local - movimiento dentro del rango PHY de las STA de comunicación, es decir, movimiento dentro de un área de servicio básica (BSA).
- b) Transición BSS: se define como un movimiento STA de un BSS en un ESS a otro BSS dentro del mismo ESS. Una transición BSS rápida es una transición BSS que establece el estado necesario para la conectividad de datos antes de la reasociación en lugar de después de la reasociación.
- c) Transición ESS: se define como el movimiento STA de un BSS en un ESS a un BSS en un ESS diferente. El mantenimiento de las conexiones de la capa superior no puede ser garantizado por el estándar IEEE 802.11; de hecho, es probable que ocurra la interrupción del servicio.



3.2 Formatos del frame MAC [16]

Cada *frame* consiste en los siguientes componentes básicos:

- a. Un encabezado MAC, el cual comprende el control de frame, duración, dirección, información del control de secuencia e información de control QoS; este último para el caso de frames de datos QoS;
- b. Un cuerpo de frame de longitud variable, el cual contiene información específica al tipo y subtipo de frame;
- c. Un FCS (*Frame Check Sequence*), el cual contiene un CRC (*Cyclic Redundancy Check*) IEEE de 32 bit.

El formato de una trama MAC contiene un grupo de campos que toman lugar en un orden fijo en todos los frames. En la figura 3.4 se muestra un formato general de una trama MAC. Los primeros tres campos (*Control de Frame*, *Duration/ID*, and *Address 1*) y el último campo de secuencia de chequeo de trama (FCS, por sus siglas en inglés) constituyen el formato de trama mínimo y está presente en todas las tramas, incluyendo los tipos y subtipos reservados. Los campos *Address 2*, *Address 3*, *Sequence Control*, *Address 4*, *QoS Control* y *Frame Body* se encuentran presentes solo en algunos tipos y subtipos [17].

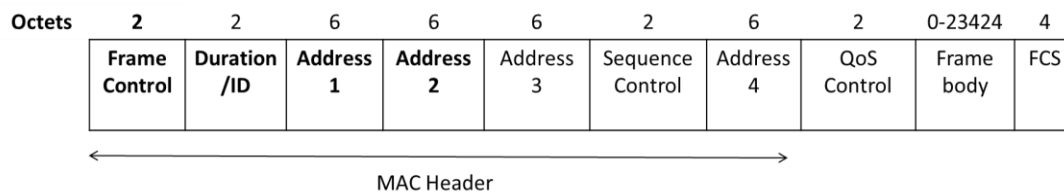


Fig 3.4 Formato de una trama MAC [17].

- Frame Control: consta de dos bytes que indican que direcciones serán incluidas en la trama.
- Duration/ID: tiene una longitud de 16 bits y su contenido varía con el tipo y subtipo de frame. Para frames de control transmite el identificador de asociación y para las tramas que no son QoS se usa el valor constante 32768. Para otros tipos de tramas, este campo contiene la duración del periodo que ha reservado el medio para una estación.
- Address 1-4: estos campos son usados para indicar el identificador del conjunto de servicios básicos (BSSID), dirección de origen (SA), dirección de destino (DA), dirección de la estación transmisora (TA) y dirección de la estación receptora (RA). Algunos usos de los campos de direcciones se especifican por la posición relativa de los campos de dirección dentro del encabezado MAC, independientemente del tipo de dirección presente en ese campo. Por ejemplo, una coincidencia de la dirección receptora siempre se lleva a cabo en el contenido del campo *Address 1* en los frames recibidos, y la dirección recibida de los frames CTS y ACK son



obtenidos siempre desde el campo *Address 2* en el frame RTS correspondiente, o desde la trama que ha sido reconocida.

- ***Sequence Control***: tiene 16 bits de longitud, realiza a manera general el control de secuencia de frames y contiene dos subcampos: *Sequence Number* y *Fragment Number*.
- ***QoS Control***: tiene 16 bits de longitud que identifica la categoría de tráfico o el flujo de tráfico (TC o TS respectivamente, por sus siglas en inglés) al cual pertenece el frame y otras informaciones relacionada con QoS sobre el frame que varían de acuerdo a su tipo y subtipo. Este campo se encuentra presente en todos los frames de datos en los cuales el subcampo QoS del campo Subtipo esté establecido en 1.
- ***Frame body***: tiene longitud variable que contiene información específica a los tipos y subtipos del frame individual. Contiene los datos de usuario.
- ***FCS***: es un campo de 32 bits que contiene un CRC de 32 bits. El FCS se calcula sobre todos los campos del encabezado MAC y el campo *Frame Body*. Para ello se utiliza un polinomio generador estándar de grado 32.

El campo de control (*Frame Control*) de 16 bits está a su vez formado por otros subcampos, los cuales se muestran en la figura 3.5.

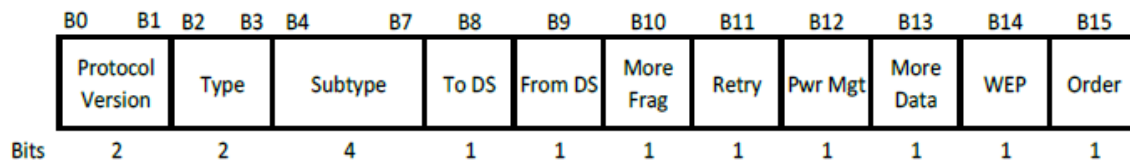


Fig 3.5 Estructura del campo Frame Control [17].

Los subcampos que conforman el campo de control son:

- ***Protocol Version***: indica la versión del estándar 802.11 y ambos bits son '0'.
- ***Type/Subtype***: identifican la función del frame, el campo *Type* indica si es un frame de datos, de control o de gestión, adicionalmente se debe especificar el subtipo de frame.
- ***To DS/From DS***: indica si la trama se envía o es recibida del sistema de distribución DS.
- ***More Fragments***: es puesto a 1 si el paquete proveniente de capas superiores ha sido fragmentado.
- ***Retry***: si el frame es una retransmisión este campo se coloca en '1'.
- ***Power Management***: activa el modo de ahorro de energía.
- ***More Data***: indica si una estación tiene más tramas que transmitir para una estación destino específica.
- ***WEP Field***: indica si la trama ha sido procesada con el algoritmo de encriptación WEP (*Wireless Equivalent Privacy*)
- ***Order Field***: es activado cuando es necesario el envío ordenado de paquetes.



Existen tres tipos de tramas MAC definidas en el estándar IEEE 802.11, éstos son:

- Gestión: tienen como objetivo establecer la comunicación entre las estaciones y el AP.
- Control: una vez se asocian y autentican la estación con un AP, estas tramas ayudan a establecer una correcta comunicación. Entre éstas se encuentran los ACK, RTS, CTS y las tramas con la función PCF.
- Datos: contienen la información a transmitir.

3.3 Estándar WAVE

Algunas de las operaciones del estándar MAC IEEE 802.11 son altas consumidoras de tiempo para ser adoptadas en IEEE 802.11p, ya que generalmente las comunicaciones vehiculares utilizan capacidades de intercambio de datos instantáneas y, por ende, no puede permitirse el escaneo de canales para detección de *beacon* en un BSS y consecuentemente ejecutar múltiples mecanismos *handshakes* para establecer la comunicación. Es esencial para los radios 802.11p estar predeterminadamente en el mismo canal y ser configurados con el mismo BSSID.

Debido a ello, se introduce en WAVE IEEE 802.11p el término “modo WAVE”. Una estación en modo WAVE tiene permitido transmitir y recibir tramas de datos con el valor de la *wildcard* BSSID y sin la necesidad de pertenecer a un BSS a priori. Por tanto, dos vehículos se pueden comunicar sin utilizar tiempo adicional tanto operen en el mismo canal utilizando la *wildcard* de la BSSID.

El estándar WAVE introduce un nuevo tipo de BSS: WBSS (WAVE BSS). Una estación forma un WBSS mediante la transmisión de un *beacon* bajo demanda. Una estación WAVE utiliza el *beacon* demandado para advertir un WAVE BSS. Tal advertencia es creada y consumida por mecanismos de capas superiores al IEEE 802.11. Éste contiene toda la información necesaria para que las estaciones receptoras entiendan los servicios ofrecidos en el WBSS con el objetivo de decidir si unirse, así como la información necesaria para configurarse a sí misma como miembro del WBSS [17]. Dicho esquema ofrece una sobreutilización extremadamente baja en la configuración de WBSSs al descartar todos los procesos de autenticación y asociación.

Algunos de los cambios realizados a nivel de MAC para la operación en ambientes WAVE son los siguientes [17]:

- Una estación en modo WAVE puede enviar y recibir tramas de datos con la *wildcard* BSSID con los campos “*To DS*” y “*From DS*” establecidos a cero, independientemente si es un miembro de un BSS WAVE.
- Un WAVE BSS (WBSS) es un tipo de BSS consistente en un grupo de estaciones cooperativa en modo WAVE que se comunican utilizando un BSSID común. Un WBSS es inicializado cuando un radio en modo WAVE envía un *beacon* WAVE, el cual incluye toda la información necesaria para que un receptor se una.



- Un radio se une a un WBSS cuando está configurado para enviar y recibir frames de datos con el BSSID definido para ese WBSS. De la misma forma, deja de pertenecer a un WBSS cuando su MAC termina de enviar y recibir tramas que usen el BSSID de ese WBSS.
- Una estación no debe pertenecer a más de un WBSS en un mismo momento. Una estación en modo WAVE no debería unirse a una infraestructura BSS o IBSS, y no debe utilizar escaneo pasivo o activo, además de que no puede utilizar procesos de autenticación o asociación MAC.
- Un WBSS deja de existir cuando no tiene miembros. El radio iniciador no difiere de ningún otro miembro después del establecimiento de un WBSS. Por consiguiente, un WBSS puede continuar si el radio iniciador deja de ser miembro.

3.4 Especificación de la capa PHY OFDM para la banda de 5GHz.

El anexo IEEE 802.11p permite el uso de la banda de frecuencia de los 5.9 GHz (5.850 – 5.925 GHz) con espaciado de canal de 20 MHz, 10 MHz y 5 MHz.

El sistema OFDM provee a la WLAN con capacidades de comunicaciones a nivel de datos de 6, 9, 12, 18, 24, 36, 48 y 54 Mbps. El sistema utiliza 52 subportadoras que son moduladas utilizando modulaciones BPSK o QPSK (*binary* o *quadrature phase shift keying*) o utilizando modulaciones 16-QAM o 64-QAM (16- or 64-*quadrature amplitude modulation*). Se usa la codificación de corrección de error con reenvío (codificación convolucional) con una razón de codificación de 1/2, 2/3 o 3/4 [18].

El sistema OFDM provee además una operación “*half-clocked*” utilizando espaciados de canal de 10 MHz con potencial de comunicación de datos de 3, 4.5, 6, 9, 12, 18, 24 y 27 Mbps. La operación “*half-clocked*” duplica el tiempo de símbolos y el tiempo CCA (*Clear channel assessment*) cuando se utiliza un canal de 10 MHz y es el ocupado en redes VANETs. El espaciado de las portadoras se reduce a la mitad (0.15625MHz) siendo el de 802.11a de 0.3125MHz y se duplica el tiempo de símbolo al doble (8us). Principalmente, se duplican todos los parámetros de tiempo de OFDM usado en las transmisiones regulares del 802.11a y en consecuencia, la razón de transmisión se reduce a la mitad.

El sistema OFDM también provee operación “*quarter-clocked*” utilizando un espaciado de canal de 5 MHz con un potencial de comunicación de datos de 1.5, 2.25, 3, 4.5, 6, 9, 12 y 13.5 Mbps. La operación “*quarter-clocked*” cuadruplica el tiempo de símbolos y los tiempos CCA.

El estándar 802.11, especifica un arreglo de 64 subportadoras, de las cuales 52 son de las útiles, asignadas desde -26 a 26. Las portadoras pilotos se encuentran en -21, -7, 7 y 21, las nulas en 0, en las 6 subportadoras inferiores y las 5 superiores del espectro.

La capa PHY OFDM consiste de dos funciones de protocolo:

- a. Una función de convergencia PHY, que adapta las capacidades del sistema dependiente del medio físico (PMD, por sus siglas en inglés) al servicio PHY.

Esta función es soportada por el PLCP (*physical layer convergence procedure*), que define un método de asignar las unidades de datos de servicio PLCP (PSDUs, por sus siglas en inglés) IEEE 802.11 a un formato de trama adecuado para enviar y recibir datos de usuario e información de gestión entre dos o más STAs utilizando el sistema PMD asociado.

- b. Un sistema PMD, cuya función define las características y métodos de transmisión y recepción de datos a través de un medio inalámbrico entre dos o más STAs, cada uno utilizando el sistema OFDM.

3.4.1 Funciones de la capa física (PHY) OFDM

La arquitectura OFDM PHY 5 GHz contiene dos subcapas (figura 3.6) [19]:

- Subcapa PLCP: se define esta subcapa de convergencia PHY para permitirle a la capa MAC operar con un mínimo de dependencia sobre la subcapa PMD. Esta función simplifica la interfaz de servicio PHY a los servicios de la capa MAC 802.11.
- Subcapa PMD: Proporciona un medio para enviar y recibir datos entre dos o más STAs. Esta cláusula se refiere a la banda de 5 GHz que utiliza la modulación OFDM.

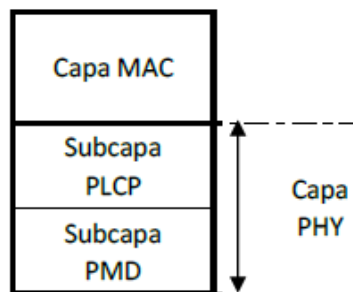


Fig 3.6 Estructura del estándar 802.11[19].

La unidad de datos de protocolo de paquetes (PPDU, por sus siglas en inglés) se compone de un preámbulo, un campo de señal y un componente de *payload* que contiene los datos útiles. El campo de preámbulo marca el comienzo del frame físico. Éste se utiliza para seleccionar la antena apropiada y corregir el desplazamiento en frecuencia y sincronía. Además, para obtener la información del estado del canal (CSI, por sus siglas en inglés), símbolos OFDM de entrenamiento o símbolos pilotos son embebidos en cada símbolo OFDM a transmitir. Éstos son utilizados para propósitos de estimación de canal y corrección de errores en la transmisión, debido a que el canal puede producir efectos indeseables en las propiedades de la señal como la alteración de la frecuencia y la fase, lo cual pueden afectar el proceso de demodulación debido a efectos como rotación de fase, corrimiento de frecuencia Doppler, degradación de la amplitud y distorsión de fase.

Todos esos efectos conllevan a una relación señal a ruido pobre y puede alterar la ubicación de la frecuencia de algunas portadoras en OFDM, lo que puede causar la pérdida de las características ortogonales de las señales. Por ende, en el transmisor se insertan símbolos bien conocidos en frecuencia, amplitud y fase, entre las subportadoras, para estimar los efectos del canal sobre la señal al momento de la demodulación y posteriormente corregirlos. El número de pilotos usados en un sistema OFDM depende de las características del canal a través del cual se envía la señal [19].

Los símbolos de entrenamiento OFDM (o preámbulo) son transmitidos al principio del proceso de transmisión, mientras que los símbolos pilotos son insertados dentro de cada símbolo OFDM, y son separados de cada símbolo de información en el dominio del tiempo [18]. La estimación del canal mediante símbolos OFDM de entrenamiento puede ser suficiente para la detección de símbolos en el caso que el canal permanezca constante sobre varios símbolos OFDM; pero en el caso de variación de canal, estos símbolos deben de ser retransmitidos frecuentemente para obtener estimaciones de canal confiables. Por otra parte, para realizar un rápido seguimiento de las variaciones del canal, se recomiendan que los símbolos pilotos sean insertados dentro de cada símbolo OFDM para facilitar la estimación de canal lo cual es conocido como: estimación de canal asistida por pilotos.

El campo de preámbulo del 802.11p (figura 3.7) está compuesto por 12 símbolos pilotos de entrenamiento los cuales se adicionan para proveer una descripción del comportamiento en frecuencia del canal y de la sincronización temporal en recepción. Éste consiste en diez símbolos de entrenamientos cortos (STS) y dos símbolos de entrenamiento largos (LTS). Siete de los 10 STS (t1-t7) son símbolos cortos OFDM los cuales son responsables de realizar un corrimiento burdo en frecuencia, sincronización temporal, permite estimación de la frecuencia de las subportadoras y la estimación de canal.

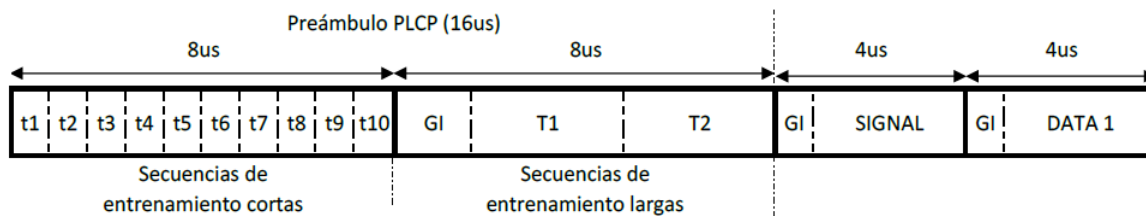


Fig 3.7 Secuencias de entrenamiento que conforman el preámbulo PLCP

La secuencia de entrenamiento corta consiste de 12 subportadoras de las 52 subportadoras de datos disponibles. De tal forma que una secuencia de entrenamiento corta normalizada en el dominio de la frecuencia está dada por:

$$S_{-26,26} = \sqrt{\frac{13}{6}} \{0,0,1 + j, 0,0,0, -1 - j, 0,0,0, 1 + j, 0,0,0, -1 - j, 0,0,0, -1 - j, 0,0,0, 1 + j, 0,0,0,0,0,0, -1 - j, 0,0,0, 1 + j, 0,0,0, 1 + j, 0,0,0, 1 + j, 0,0,0, q + j, 0,0\} \quad (3.1)$$



Las dos secuencias de entrenamiento largas son utilizadas para estimación de canal y una adquisición más fina en frecuencia en el receptor.

La secuencia de entrenamiento larga se puede definir en frecuencia como:

$$S_{-26,26} = \{1,1, -1, -1,1,1, -1,1, -1,1,1,1,1,1,1, -1, -1,1,1, -1,1, -1,1,1,1,1,0,1, -1, -1,1,1, -1,1, -1,1, -1, -1, -1, -1, -1,1,1, -1, -1,1, -1,1, -1,1,1,1,1\} \quad (3.2)$$

3.4.2 Subcapa PLCP OFDM

La subcapa PLCP se encarga de convertir los PSDUs a PPDUs y viceversa. Durante la transmisión los PSDUs deben ser provistos con un encabezado y un preámbulo PLCP para crear el PPDU. En el receptor, el preámbulo y el encabezado PLCP son procesados para ayudar en la demodulación y entrega de la PSDU.

En la figura 3.8 se muestra el formato para el PPDU incluyendo el encabezado y preámbulo PLCP OFDM, el PSDU, los bits de cola y los bits de relleno. El encabezado PLCP contiene los siguientes campos: *Length*, *Rate*, un bit reservado y uno de paridad (con un apéndice de 6 bits de cola de relleno) constituyen un solo símbolo OFDM separado denotado '*Signal*', el cual es transmitido con la combinación más robusta de la modulación BPSK y una razón de codificación de 1/2. El campo *Service* del encabezado PLCP y el PSDU (con un apéndice también de 6 bits de cola y bits de relleno) se denota como '*Data*', y son transmitidos a una razón de datos definida en el campo *Field* y puede ser constituida por múltiples símbolos OFDM. Los bits de cola en el símbolo '*Signal*' permiten la decodificación de los campos *Rate* y *Length* inmediatamente después de la recepción de los bits de cola. Estos campos son necesarios para la decodificación de la parte *Data* del paquete. Además, el mecanismo de CCA puede aumentarse prediciendo la duración del paquete desde el contenido de los campos *Rate* y *Length*, incluso si la razón de datos no es soportada por el STA.

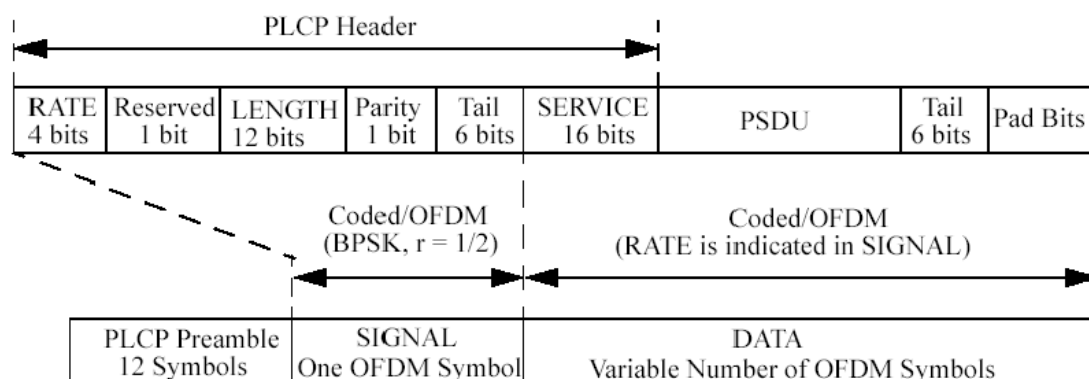


Fig 3.8 Formato de trama PPDU [16].



Como se mencionó anteriormente, el campo *SIGNAL* se compone de [16]:

- ✓ *RATE*: el cual indica la velocidad a la cual se transmiten los datos. Al indicar la tasa de transmisión, ya se puede definir el tipo de modulación y la tasa de codificación convolucional ocupada como se muestra en la tabla.
- ✓ *RESERVED*, *SIGNAL TAIL*: contiene valor ceros.
- ✓ *LENGTH*: ocupa 12 bits e indica la longitud en bytes de la trama PSDU.
- ✓ *DATA*: los símbolos de datos OFDM están compuestos a su vez de los siguientes campos:
 - *SERVICE*: contiene 16 bits en "0" y los primeros 7 son utilizados para inicializar el randomizador en el receptor. Los restantes son de uso reservado.
 - PSDU: consiste en la trama proveniente de la capa MAC.
 - *Tail*: se utiliza para reiniciar el codificador convolucional a cero. Contiene 6 bits.
 - *Pad bits*: constituye los bits de relleno para que el número total de bits de datos sea múltiplo entero del número de bits en un símbolo OFDM.

3.4.3 Proceso de codificación del PPDU

A continuación, se muestra una visión general que facilitará la comprensión del procedimiento de convergencia [16]:

- a. Producir el campo PLCP *Preamble*, compuesto de 10 repeticiones de una "secuencia de entrenamiento corta" (utilizada para la convergencia, el control automático de ganancia (AGC, por sus siglas en inglés), selección de diversidad, adquisición de temporización y adquisición aproximada de frecuencia en el receptor) y dos repeticiones de una "secuencia de entrenamiento larga" (utilizada para la estimación de canal y una adquisición fina de frecuencias en el receptor), antecedida por un intervalo de guarda (GI, por sus siglas en inglés).
- b. Producir el campo del PLCP *Header* desde los campos *RATE*, *LENGTH* y *SERVICE* del *TXVECTOR*¹ mediante el apropiado llenado. Los campos *RATE* and *LENGTH* del encabezado PLCP se codifican con un código convolucional a razón de $\frac{1}{2}$, y son mapeados secuencialmente dentro de un solo símbolo OFDM codificado con BPSK denotado como símbolo *SIGNAL*. Para facilitar una detección sincronizada y confiable de los campos *RATE* y *LENGTH* se insertan 6 bits '0' de cola dentro del encabezado PLCP. La codificación del campo *SIGNAL* dentro de un símbolo OFDM sigue los mismos pasos para la codificación convolucional, el *interleaving*, la modulación BPSK, inserción de símbolos pilotos, transformada de Fourier y anteponiendo un

¹ El *TXVECTOR* representa una lista de parámetros que la subcapa MAC provee a la entidad PHY local para transmitir un PSDU. El vector contiene parámetros de gestión PHY y PLPC.



- GI como se describe subsecuentemente para la transmisión de datos modulados OFDM-BPSK a una razón de codificación $\frac{1}{2}$. Los contenidos del campo *SIGNAL* no son mezclados.
- Calcular desde el campo *RATE* del *TXVECTOR* el número de bits de datos por símbolo OFDM (N_{DBPS}), la razón de codificación (R), el número de bits por cada subportadora OFDM (N_{BPSC}), y el número de bits codificados por símbolo OFDM (N_{CBPS}).
 - Agregar el PSDU al campo *SERVICE*. Se extiende la cadena de bits resultante con bits '0' (al menos 6 bits), por ende, la longitud resultante será un múltiplo de N_{dbps} . La cadena de bits resultantes constituye la parte *DATA* del paquete.
 - Se inicia el *scrambler* con una semilla pseudo-aleatoria no nula, se genera una secuencia de *scrambling* y se realiza una XOR con la cadena extendida de bits de datos.
 - Se reemplazan los seis bits ceros codificados que siguen a los datos con seis bits '0' *non-scrambled*. (estos bits regresan el codificador convolucional al estado cero y se indican como bits de cola).
 - Codificar la cadena de datos ya mezclada y extendida con un código convolucional $R=1/2$.
 - Dividir la cadena de bits codificada en grupos de N_{CBPS} bits. Dentro de cada grupo, se realiza un *interleaving* (reorganización) de los bits de acuerdo a las reglas correspondientes de la razón deseada.
 - Dividir el código resultante y la cadena de datos reorganizada en grupos de N_{CBPS} bits. Para cada uno de los grupos de bits, se convierte el grupo de bits a un número complejo de acuerdo a las tablas de codificación y modulación.
 - Dividir la cadena de números complejos en grupos de 48 números complejos. Cada grupo estará asociado con un símbolo OFDM. En cada grupo, los números complejos serán numerados desde 0 a 47 y mapeados dentro de las portadoras OFDM numeradas desde -26 a -22, -20 a -8, -6 a -1, 1 a 6, 8 a 20 y 22 a 26. Las subportadoras -21, -7, 7 y 21 son omitidas y subsecuentemente, utilizadas para la inserción de subportadoras pilotos. La subportadora 0, asociada con la frecuencia central, es omitida y rellenada con valor cero.
 - Insertar las cuatro subportadoras pilotos en las posiciones -21, -7, 7 y 21. El número total de las subportadoras es 52 (48 + 4).
 - Por cada grupo de subportadoras -26 a 26, se convierten las subportadoras al dominio del tiempo utilizando la transformada inversa de Fourier (IFFT).
 - Anexar a los símbolos OFDM uno después del otro, comenzando después del campo *SIGNAL* que describe los campos *RATE* y *LENGTH*.
 - Convertir la forma de onda compleja en banda base a RF de acuerdo a la frecuencia central del canal deseado y realizar la transmisión.

Los parámetros dependientes de la modulación sobre la razón de datos deberían ser establecidos de acuerdo a la siguiente tabla:



Tabla 3.1 Parámetros dependientes de la tasa de transmisión [16]

Esquema de modulación	Razón de codificación (R)	Bits codificados por subportadora N_{BPSC}	Bits codificados por símbolo OFDM N_{CBPS}	Bits de datos por símbolo OFDM N_{DBPS}	Razón de transferencia (Mbps) (espaciamiento de canal 20 MHz)	Razón de transferencia (Mbps) (espaciamiento de canal 10 MHz)	Razón de transferencia (Mbps) (espaciamiento de canal 5 MHz)
BPSK	1/2	1	48	24	6	3	1.5
BPSK	3/4	1	48	36	9	4.5	2.25
QPSK	1/2	2	96	48	12	6	3
QPSK	3/4	2	96	72	18	9	4.5
16-QAM	1/2	4	192	96	24	12	6
16-QAM	3/4	4	192	144	36	18	9
64-QAM	2/3	6	288	192	48	24	12
64-QAM	3/4	6	288	216	54	27	13.5

3.5 MIMO (*Multiple Input Multiple Output*)

En telecomunicaciones, *Multiple Input Multiple Output* (múltiples entradas múltiples salidas) se refiere al uso de más de una antena en el transmisor y/o en el receptor. MIMO se ha convertido a lo largo de los últimos años en una alternativa más que atractiva para el desarrollo de los medios de transmisión que abarcan tanto el medio inalámbrico, como los sistemas de fibra óptica basados en transmisión coherente. De esta forma, ha pasado a formar parte de los estándares de diversas e importantes tecnologías donde se pueden distinguir LTE (*Long Term Evolution*) o Wi-Fi entre otras. Además, es de suponer una constante evolución en los próximos años que haga de ella una opción incluso mejor para incorporarse progresivamente a más tipos de redes.

MIMO ofrece un incremento significativo en la razón de transferencia de datos y en el alcance del enlace sin una necesidad adicional de ancho de banda o potencia de transmisión. Esta tecnología contribuye a mitigar la degradación que sufre la señal sobre el canal inalámbrico y las limitaciones de recursos. La ganancia en el desempeño resultante de sistemas MIMO son ganancia de arreglos, ganancia de diversidad espacial, ganancia de multiplexación espacial y reducción de interferencia [14, 20].

- Una ganancia de arreglos significa una ganancia de potencia de las señales transmitidas logrado mediante el uso de múltiples antenas en el transmisor y/o receptor, con respecto al caso SISO (*single-input single-output*). Ello resulta en una ganancia de potencia debido a un efecto de combinación coherente de la señal inalámbrica en el receptor mejorando así la resistencia al ruido y, por tanto, conlleva a una mejora de la cobertura en la red inalámbrica.
- La ganancia de diversidad espacial se refiere a la habilidad para mitigar el desvanecimiento mediante el uso de múltiples antenas para mejorar la calidad y la confiabilidad de un enlace inalámbrico. Esta ganancia surge debido a que los múltiples enlaces creados por MIMO ofrecen una visión diversa en el receptor de la misma señal transmitida.



- c) El multiplexado espacial es un caso especial de MIMO que transmite señales de datos independientes denominados 'flujos'. De esta forma, la dimensión espacial es reutilizada o multiplexada más de una vez. La ganancia por multiplexado espacial se refiere al incremento de la razón de datos debido al uso de MIMO.
- d) La reducción de interferencia se refiere a la habilidad de MIMO para explotar la dimensión espacial e incrementar la separación entre usuarios, los cuales comparten los recursos de tiempo y frecuencia.

Sin embargo, las investigaciones y el estándar para VANETs, IEEE 802.11p, no incluye especificaciones para una versión MIMO, pero debido a que el canal de radio en VANETs es muy variable, se supone que los sistemas MIMO pueden ser muy prometedores para las redes vehiculares [21].

En el contexto de sistemas MIMO, se pueden implementar varias versiones en cuanto a en cual lado (transmisión o receptor) existen múltiples entradas o salidas. Debido a ello, se describen los siguientes tipos de sistemas MIMO [21]:

- *Single input, single output (SISO)*: es el Sistema clásico, sin capacidad MIMO, tal como el propuesto en el estándar IEEE 802.11p.
- *Single input, multiple outputs (SIMO)*: es denominado diversidad en recepción, en cuyo lado se utilizan más de una antena.
- *Multiple inputs, single output (MISO)*: se denomina diversidad en transmisión, en cuyo lado se utiliza más de una antena.
- *Multiple inputs, multiple outputs (MIMO)*: significa una diversidad completa en transmisión y recepción.

3.5.1 Codificación espacio-temporal

La codificación espacio-temporal constituye una serie de técnicas que son utilizadas en las comunicaciones inalámbricas para transmitir múltiples copias de una cadena de datos a través de un número de antenas y explotar las diferentes versiones recibidas de los datos para mejorar la confiabilidad de transferencia de los mismos. El hecho de que la señal transmitida deba atravesar un ambiente potencialmente difícil con dispersión, reflexión, refracción, etc., y pueda además corromperse por el ruido térmico en el receptor, conlleva a que algunas de las copias recibidas de los datos sean "mejores" que otras. Esta redundancia resulta en una mejor oportunidad de ser capaz de utilizar una o más de una de las copias de la señal recibida para decodificar correctamente la señal transmitida. La codificación espacio-temporal combina las copias de la señal recibida de manera óptima para extraer tanta información de cada una de ella como sea posible.

Existen varias investigaciones en el área de la codificación espacio-temporal. En la presente investigación se ocupa el código espacio-temporal propuesto por Alamouti [22] en 1998. Este esquema es un esquema de diversidad en transmisión diseñado para dos antenas transmisoras y M antenas receptoras, donde el orden de diversidad es de 2M.

3.5.2 Esquema Alamouti

El esquema Alamouti [22] puede utilizar diversidad espacial y temporal en la transmisión y es considerado uno de los esquemas más simples de diversidad espacio-temporal a ser implementado en sistemas inalámbricos MIMO. En la figura 3.9 se presenta el sistema MISO 2x1 propuesto por Alamouti.

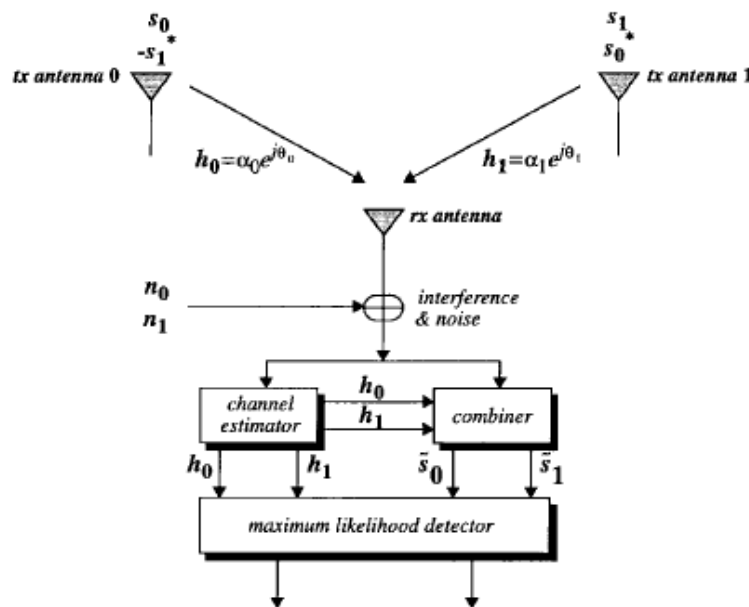


Fig 3.9 Esquema de diversidad con dos antenas en transmisión y una en recepción [22].

En un periodo de símbolo dado, dos símbolos de datos son transmitidos simultáneamente desde las dos antenas. El símbolo transmitido desde la antena 0 y antena 1 se denota como S_0 y S_1 respectivamente. Durante el próximo período de símbolo ($t + T$), se transmite la señal $-S_1^*$ desde la antena 0 y la señal S_0^* desde la antena 1, donde ' T ' es la duración de símbolo y '*' representa el complejo conjugado del número. En la tabla 3.2 se muestra la secuencia de codificación en espacio y tiempo (codificación espacio-temporal). Sin embargo, también la codificación puede ser llevada a cabo en espacio y frecuencia. En vez de dos periodos de símbolos adyacentes, se pueden utilizar dos portadoras adyacentes (codificación espacio-frecuencia).



Tabla 3.2 Secuencia de codificación y transmisión para un esquema de diversidad de dos antenas transmisoras.

Tiempo	Antena 0	Antena 1
t	s_0	s_1
$t + T$	$-s_1^*$	s_0^*

El canal en un tiempo t puede ser modelado por una distorsión multiplicativa compleja $h_0(t)$ para la transmisión por la antena 0 y $h_1(t)$ para la transmisión por la antena 1. Asumiendo que el desvanecimiento es constante a través de dos símbolos consecutivos, se establece que [22]:

$$h_0(t) = h_0(t + T) = h_0 = \alpha_0 e^{j\theta_0} \quad (3.1)$$

$$h_1(t) = h_1(t + T) = h_1 = \alpha_1 e^{j\theta_1} \quad (3.2)$$

Donde 'T' es la duración de símbolo y α_i y θ_i denotan la ganancia en amplitud y la rotación de fase sobre los dos símbolos. Por ende, la señal recibida puede ser expresada como:

$$r_0 = r(t) = h_0 s_0 + h_1 s_1 + n_0 \quad (3.3)$$

$$r_1 = r(t + T) = -h_0 s_1^* + h_1 s_0^* + n_1 \quad (3.4)$$

Donde r_0 y r_1 son las señales recibidas en el tiempo t y $t+T$ respectivamente y n_0 y n_1 son variables aleatorias complejas representando el ruido y la interferencia.

El combinador que se presenta en la figura 3.9, construye las dos señales combinadas que son enviadas al detector de máxima verosimilitud según la modulación utilizada:

$$s_0 = h_0^* r_0 + h_1 r_1^* \quad (3.5)$$

$$s_1 = h_1^* r_0 - h_0 r_1^* \quad (3.6)$$

Las principales aportaciones de este esquema son que I) la tasa del código es 1 dado que se transmiten dos símbolos en dos intervalos y II) no requiere de un canal de realimentación para conseguir diversidad en transmisión.

3.6 Estimación del canal

En la figura 3.10 se muestra un diagrama de bloques simplificado de codificación espacio-temporal de un sistema OFDM con dos antenas transmisoras y una antena receptora. Como se observa en la misma, se requiere una correcta estimación del canal para decodificar la señal recibida en el esquema de Alamouti. La estimación de canal abarca los dos canales tomados por la señal: 1) desde antena 0 al receptor y 2) desde antena 1 hacia el receptor.

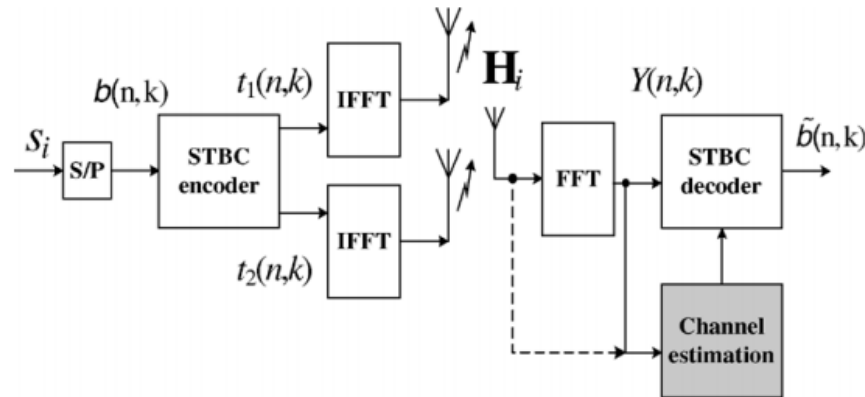


Fig 3.10 Sistema con codificación espacio-temporal OFDM con dos antenas transmisoras

Para realizar una correcta decodificación del código espacio-temporal se tiene que realizar una estimación de los parámetros del canal de manera prácticamente exacta. Por lo que dicho tópico constituye un punto fundamental dentro de los sistemas MIMO que utilicen cierta codificación.

Para obtener la información del estado del canal (CSI, por sus siglas en inglés) se utilizan símbolos OFDM de entrenamiento o símbolos pilotos embebidos en cada símbolo OFDM. Los símbolos OFDM de entrenamiento o el preámbulo OFDM son transmitidos al comienzo de la transmisión, mientras que los símbolos pilotos son embebidos dentro de cada símbolo OFDM y son separados de símbolos de información en el dominio de la frecuencia [23, 24].

Si el canal permanece constante sobre varios símbolos OFDM, la estimación del canal mediante el preámbulo OFDM puede ser suficiente para la detección del símbolo; pero si el canal presenta eventos de variabilidad en el tiempo, se requieren enviar los símbolos de entrenamiento OFDM de manera frecuente para obtener una estimación confiable del canal. Por otra parte, para realizar un seguimiento de un canal variable rápidamente se deben de insertar los símbolos pilotos dentro de cada símbolo OFDM para facilitar la estimación del canal. Esta técnica se denomina estimación de canal asistida por pilotos (*pilot-assisted* o *pilot-aided*) [25, 26].

3.6.1 Estimación del canal asistido por pilotos

La estimación del canal insertando los pilotos dentro de todas las subportadoras de un símbolo OFDM con un periodo específico, se denomina estimación de canal basada en distribución de pilotos en modo bloque y se muestra en la figura 3.11 [27]. Con esta distribución, los símbolos utilizados para la estimación de canal se transmiten periódicamente en un símbolo OFDM con todas las subportadoras usadas como pilotos. Si el canal es constante durante todo el bloque, y en ausencia de ruido, el error de estimación del canal es nulo puesto que el valor de todas las subportadoras disponibles es conocido. En este arreglo de pilotos tipo bloque se utiliza generalmente el algoritmo denominado *Least Square*

(LS, por sus siglas en inglés) por su simplicidad de procesamiento. El mismo será tratado más adelante en este apartado.

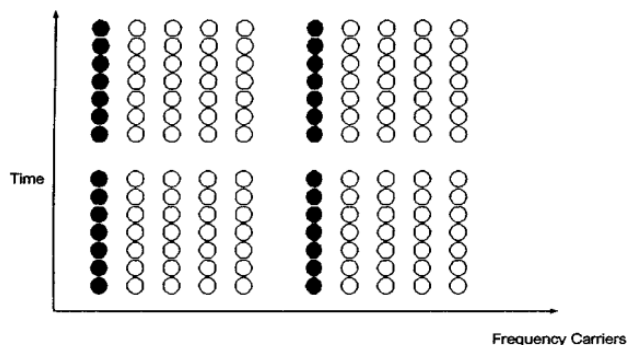


Fig 3.11 Arreglo de pilotos tipo bloque [27].

Por su parte, en la estimación de canal basada en distribución de pilotos en modo combinado (*comb-type pilots*), un número $N_p < N$ de tonos piloto son insertados uniformemente dentro de cada símbolo OFDM, de manera que las $N - N_p$ subportadoras restantes envían información. Así se consigue en equilibrio entre capacidad de estimación del canal y eficiencia espectral. La misma fue desarrollada para satisfacer la necesidad de ecualizar cuando el canal cambia en cada bloque OFDM. La estimación del canal con pilotos ubicados en modo combinado consiste en realizar una estimación en las frecuencias pilotos e interpolar el canal [27].

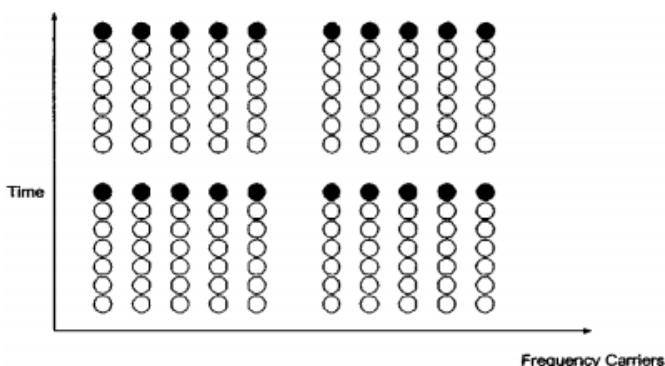


Fig 3.12 Arreglo de pilotos tipo combinado [27].

La distribución en modo combinado introduce una moderada ineficiencia espectral (del orden de $1 - N_p/N$) para proporcionar capacidad de seguimiento de canales variantes en el tiempo. Sin embargo, la calidad de la estimación del canal está en conflicto con dicha capacidad de seguimiento, es decir, a mayor densidad de tonos piloto, peor eficiencia espectral.



Como se mencionó anteriormente, el estándar 802.11p requiere cuatro subportadoras pilotos que son utilizadas para hacer un seguimiento del corrimiento de frecuencia residual entre el transmisor y el receptor. La estimación del canal que comprende el estándar se realiza mediante el envío de dos símbolos de entrenamiento idénticos al comienzo del paquete. Dicho método presenta el inconveniente de que el canal puede variar considerablemente durante el tiempo en que se transmite un paquete.

En el presente trabajo de investigación se explicarán dos estimadores de canal: mínimos cuadrados o LS e Interpolación de pilotos distribuidos de forma combinada. El primero de éstos es el estimador de canal implementado tradicionalmente en equipos 802.11 y el segundo, es el que ocuparemos en el presente proyecto.

a) Estimación de mínimos cuadrados (LS)

Recordando lo expuesto en la sección 3.4.1, la estructura del paquete 802.11p contiene 10 secuencias de entrenamiento cortas ($t_1 - t_{10}$), usadas para sincronización y dos secuencias de entrenamiento largas ($T_1 - T_2$) que son idénticas y son utilizadas para estimar el canal.

El estimador LS [28], es uno de los más simples y utiliza solo el preámbulo en el comienzo de un paquete (secuencia de entrenamiento larga) para estimar el canal y subsecuentemente, con ese coeficiente estimado ecualiza los símbolos futuros. Este ecualizador básico utiliza, además, los pilotos en el paquete para corregir el corrimiento residual en frecuencia, lo cual es un efecto colateral de no realizar una estimación perfecta en el corrimiento en frecuencia desde el preámbulo de entrenamiento. Este tipo de ecualización representa el tipo más básico de su tipo, es utilizada generalmente como línea de referencia en las investigaciones, se ocupa en las tarjetas comerciales 802.11a y tiene como ventaja una gran facilidad de implementación en hardware [29].

Denotando la estimación de un valor X como \hat{X} , se calcula el canal H en la subportadora k como:

$$\hat{H}(k) = \frac{Y_1(k) + Y_2(k)}{2X_{LT}(k)} \quad (3.7)$$

Donde $Y_{1,2}$ son las dos copias de la señal recibida de la secuencia de entrenamiento larga y X_{LT} es su valor conocido. Con el ecualizador LS, la estimación inicial se mantiene durante toda la trama [30].

b) Estimación con Interpolación de pilotos distribuidos de forma combinada

Este tipo de ecualizador interpola linealmente en el dominio de la frecuencia los cuatro símbolos pilotos que son enviados de manera intercalada con los símbolos de datos. En el estándar 802.11p los pilotos usados para la corrección en corrimiento de frecuencia son espaciados por 14 subportadoras (2.1875 MHz). En este esquema, como primer paso se demodula cada símbolo utilizando la Transformada Discreta de Fourier (DFT, por sus siglas en inglés) como se explica en [29]. Luego, se extraen los valores recibidos en el dominio de la frecuencia de las subportadoras pilotos y se designan a un vector Y_p de 4 elementos. Los valores DFT conocidos de los pilotos enviados en esas subportadoras, los



especificados por el estándar, forman otro vector X_p de cuatro elementos. La estimación LS en cada subportadora piloto está dada por:

$$H(k) = \frac{Y_p(k)}{X_p(k)} \quad (3.8)$$

Lo anterior define un vector de cuatro elementos que representa las estimaciones espaciadas uniformemente del canal. Para interpolar las estimaciones, los primeros extremos son anexados al vector para obtener:

$$H'_p = [m_{H_p} \ H_p^T \ m_{H_p}]^T \quad (3.9)$$

Donde m_{H_p} es la media de H_p . La media es usada en los extremos en lugar de la extrapolación de las subportadoras -21 a la 21 tanto por simplicidad como porque no existe forma de determinar la respuesta actual del canal a los extremos de las frecuencias. Posteriormente, H'_p se pasa a través de un circuito de interpolación, el cual ubica los ceros $L-1$ entre las muestras sucesivas en H'_p y pasa la señal resultante a través de un filtro paso bajo con frecuencia de corte π/L , donde L es 14. La salida de la operación de interpolación se recorta adecuadamente a ambos lados para producir una estimación de canal interpolada para las 53 subportadoras, designada H_{update} [29].

Después de estimar el canal en un símbolo dado, la estimación global del canal se actualiza apropiadamente para realizar un rastreo del mismo. La nueva estimación en un tiempo t está dada por la ecuación 3.10, donde α constituye un parámetro de memoria. A valor mayor de α implica mayor memoria.

$$H_t = \left(1 - \frac{1}{\alpha}\right) H_{t-1} + \frac{1}{\alpha} (H_{update}) \quad (3.10)$$

En la investigación realizada en [30] se comparan ambos algoritmos de estimación de canal: el LS con respecto a la secuencia de entrenamiento larga y el de Interpolación de pilotos distribuidos de forma combinada se utiliza tomando como referencia las portadoras pilotos que se encuentran entre las portadoras de datos. Debido a que este último logró mejor desempeño en cuanto al porcentaje de tramas recibidas tanto en diferentes entornos, con diferentes tamaños de tramas y a diferentes velocidades, se decide ocupar en la presente investigación.



CAPÍTULO 4. *Software Defined Radio* (SDR)

El término “*Software Defined Radio*” (SDR) se define como la realización en software del procesamiento y la manipulación de todas las señales empleadas en las comunicaciones de radio, en vez de utilizar hardware [31].

El presente capítulo se divide en dos secciones principales. En un primer momento se pretende dar una introducción y definir las características de la plataforma SDR ya que constituye un concepto primordial en la implementación de la tesis.

En un segundo momento se describirán detalladamente las plataformas SDR utilizadas para la implementación de este proyecto, así como su arquitectura y características.

4.1 Introducción a la plataforma SDR

Una radio definida por software (SDR) es un dispositivo inalámbrico que se compone normalmente de un extremo de RF configurable con una FPGA o un system-on-chip (SoC) programable para realizar funciones digitales. El hardware SDR disponible comercialmente puede transmitir y recibir señales en frecuencias distintas para implementar los estándares inalámbricos, desde radio FM a Wi-Fi y LTE. Los ingenieros de tecnologías inalámbricas pueden usar el hardware de radio definida por software como una plataforma rentable y en tiempo real para una amplia variedad de tareas de ingeniería inalámbrica, tales como [32]:

- Realización de pruebas en laboratorio y pruebas de campo con señales de RF en vivo
- Creación rápida de prototipos de funciones de radio personalizadas
- Aprendizaje práctico de conceptos y habilidades de diseño relacionados con las comunicaciones inalámbricas

El concepto SDR fue introducido por Joseph Mitola en 1991 [1] para referirse a “radios” reconfigurable; es decir, un SDR es un dispositivo de comunicaciones cuya funcionalidad se encuentra definida en el software, lo cual minimiza la necesidad de realizar modificaciones de hardware durante actualizaciones tecnológicas. Por ende, el desarrollo de SDR sobrepone las desventajas del hardware de radio ya que de manera general, la mayoría de los dispositivos son muy caros, además de sus componentes tales como: filtros, moduladores, demoduladores y conversores para el manejo de las frecuencias.

La principal diferencia entre SDR y radio de hardware es la utilización de un usuario para analizar las bandas de frecuencias disponibles e integrar diferentes tecnologías de conexión en una sola interfaz. Esto significa que un usuario sólo puede utilizar un dispositivo para recibir señales WiMAX, WiFi, GSM o LTE. Aunque cada señal tiene diferentes métodos de codificación y análisis, se requiere descargar los módulos estandarizados más recientes, tal como importar una librería a los compiladores generales, etc., con el fin de recibir e interpretar la radio de manera normal y lograr una reducción de costes y ahorro de espacio. Desde una perspectiva de gestión, la administración de radio de hardware es flexible debido a que éste sólo puede administrar conexiones, y cambiar parámetros de TCP/IP y de aplicaciones. Por el contrario, el rango de gestión de SDR cubre la capa física [33].

De manera general se pueden mencionar ventajas de la tecnología SDR tales como su alto nivel de reconfiguración permitiendo una coexistencia común de distintos estándares mediante una configuración dinámica, su interoperabilidad facilitando la implementación de sistemas con arquitectura abierta y su confiabilidad al establecer la opción de implementar e innovar con múltiples algoritmos de prevención, detección y corrección de errores [34].

Como se muestra en la figura 4.1, de manera general un SDR se divide en:

- *RF Front-End*: sirve de interface al dominio RF analógico, es el encargado de enviar y recibir las señales en radiofrecuencia a través de una antena. Se conforma de filtros, amplificadores de bajo ruido, amplificadores de potencia, osciladores locales y realiza la conversión de RF a frecuencia intermedia.
- Procesador Digital de Señales (DSP): se implementa la parte de inteligencia en hardware formando la interfaz entre el dominio analógico y el digital, incluyendo la conversión entre los mismos. Generalmente se utilizan *FPGAs Field-programmable gate array*.
- CPU o unidad de procesamiento central: se realiza todo el procesamiento de la señal y se implementa completamente en software.

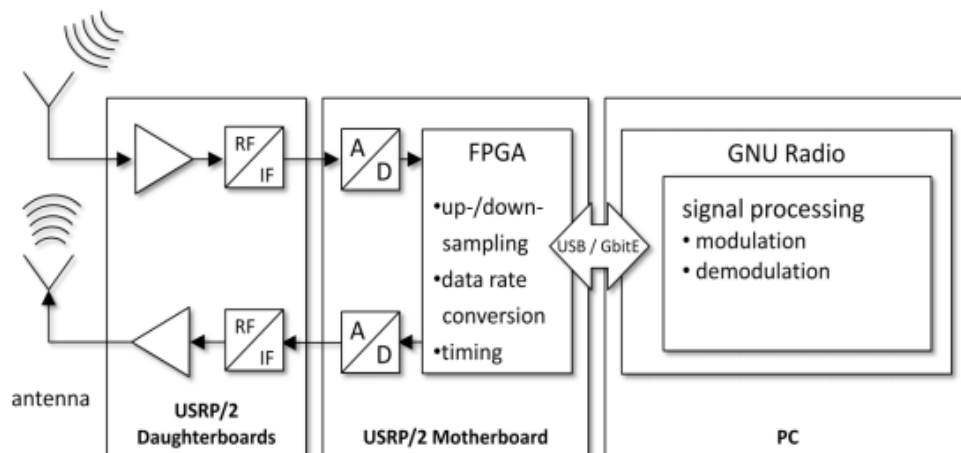


Fig 4.1 Diagrama en bloques de SDR [35].



4.2 Universal Software Radio Peripheral (USRP)

Universal Software Radio Peripheral (USRP) es una gama de radios definidos por software (SDR) diseñados y vendidos por *Ettus Research* y su empresa matriz, *National Instruments*. La familia de productos USRP pretende ser una plataforma de radio software relativamente económica, desarrollada por Matt Ettus [35] y es usada ampliamente en investigaciones y ambientes educativos.

La mayoría de los USRP se conectan a una computadora a través de un enlace de alta velocidad en la que se encuentra el software utilizado para controlar el hardware USRP y los datos que se transmiten o reciben. También existen algunos modelos de USRP (como la serie embebida E de Ettus) que integran funcionalidades de procesamiento en software para operar en un modo *Stand-alone*, aunque la arquitectura utilizada es similar de manera general.

El USRP se encuentra formado por dos módulos generales denominados: tarjeta madre (*motherboard*) y el *front-end* que constituye la tarjeta hija (*daughterboard*). Una tarjeta madre provee subsistemas para el procesamiento de la señal en banda base como: generación de reloj y sincronización, una *Field-programmable gate array* (FPGA), ADCs, DACs, interfaz de procesamiento y regulación de potencia. La tarjeta hija se encarga de realizar operaciones analógicas, conversión y filtrado como se mencionó anteriormente.

La comunicación entre el USRP y el procesador central localizado generalmente en una computadora, se realiza a través del *USRP Hardware Driver* (UHD). Éste es provisto por Ettus Research para ser utilizado con la familia de productos USRP, es compatible con sistemas operativos Linux, MacOS y Windows.

Para el desarrollo del presente proyecto se utilizan dos modelos de plataformas SDR comerciales de la compañía Ettus Research, el USRP N210 y el USRP X310.

4.2.1 USRP N210

El *Universal Software Radio Peripheral* (USRP) N210, cuya hoja de datos se muestra en el anexo A, es una plataforma flexible de bajo costo que opera con la ayuda de una computadora de propósito general y provee un rango amplio de ancho de banda además de una alta capacidad de procesamiento.

El USRP N210 se considera para aplicaciones demandantes de comunicaciones que requieran este tipo de rápido desarrollo. La arquitectura incluye una FPGA Xilinx® Spartan® 3ª-DSP 3400, ADC dual de 100 MS/s/-14 bits, DAC dual 400 Ms/s-16 bit y conectividad *Gigabit Ethernet* para transportar el flujo de datos desde y hacia el procesador. El diseño modular permite al USRP N210 operar desde DC hasta los 6 GHz, mientras que un puerto de expansión permite a múltiples dispositivos de la serie N210 estar sincronizados y ser utilizados en una configuración MIMO. Un módulo GPSDO opcional se puede usar también para controlar el reloj de referencia a 0.01 ppm del estándar GPS global. La arquitectura del dispositivo se muestra en la figura 4.2.

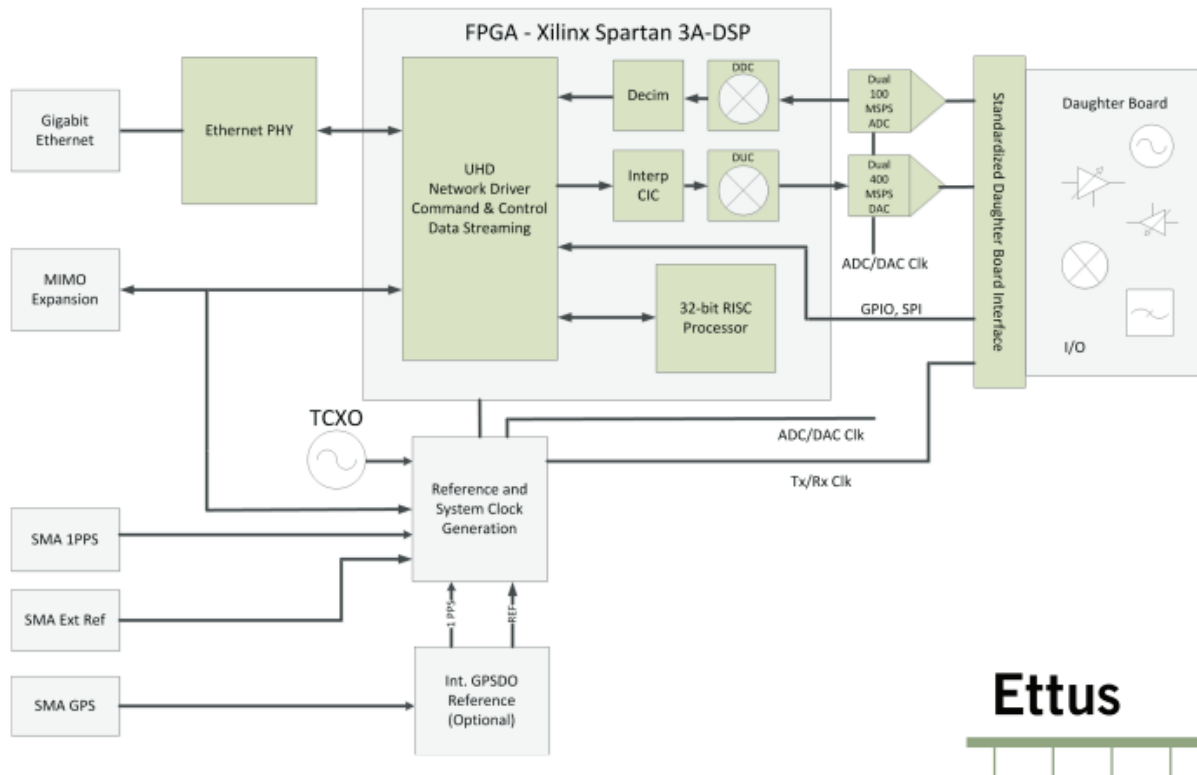


Fig 4.2 Arquitectura de la USRP N210.

El USRP N210 puede transportar un máximo de 50 MS/s desde y hacia la aplicación de software simultáneamente (modo full-dúplex), además los usuarios pueden implementar funciones personalizadas directamente en la FPGA del dispositivo. La FPGA ofrece el potencial de procesar un máximo de 100 MS/s tanto en recepción como en transmisión, y su *firmware* puede ser cargado a través de la interfaz Gigabit Ethernet [36].

Otras especificaciones del dispositivo se muestran en la tabla 4.1.

Tabla 4.1 Especificaciones del USRP N210 [37].

Especificación	Tipo	Unidad
Potencia		
Alimentación DC	6	V
Consumo de corriente	1.3	A
Relojes y desempeño de conversión		
Razón de muestreo ADC	100	MS/s
Resolución ADC	14	Bits
Razón de muestreo DAC	400	MS/s
Resolución DAC	16	bits
Razón de muestreo de <i>host</i> (8b/16b)	50/25	MS/s



Exactitud en frecuencia	2.5	ppm
Referencia GPSDO	0.01	ppm
Física		
Temperatura operación	0 a 55°	C
Dimensiones (l x w x h)	22 x 16 x 5	cm
Peso	1.2	kg

4.2.2 USRP X310

El USRP X310 de Ettus Research (Anexo B) es una plataforma SDR escalable y de alto desempeño para el diseño y despliegue de sistemas de comunicaciones inalámbricos de próxima generación. La arquitectura de hardware combina dos ranuras de tarjetas hijas de ancho extendido cubriendo desde DC a 6 GHz con un máximo de hasta 160 MHz de ancho de banda en banda base, con múltiples opciones de interfaces de alta velocidad como son: PCIe, dual 10 GigE y dual 1 GigE, además de una potente FPGA Kintex-7 (XC7K410T) programable para el DSP de alto desempeño. La misma provee una conectividad de alta velocidad entre todos los componentes dentro del dispositivo incluyendo el *front-end* de radio, las interfaces del usuario, y la memoria DDR3. Posee una arquitectura de software de código abierto y provee soporte para el driver UHD lo cual la hace compatible con un gran número de esquemas de desarrollos, arquitecturas de referencias y proyectos de código abierto.

Las interfaces de alta velocidad que posee son las siguientes [38]:

- Dual 10 Gigabit Ethernet - 2x RX at 200 MSps per channel
- Dual 10 Gigabit Ethernet - 4x RX at 80 MSps per channel
- PCIe Express (Desktop) - 200 MS/s Full Duplex
- ExpressCard (Laptop) - 50 MS/s Full Duplex
- Dual 1 Gigabit Ethernet - 25 MS/s Full Duplex

Se recomienda que para aplicaciones que necesiten grandes anchos de bandas y baja latencia como investigaciones en capas PHY/MAC, se utilice la interfaz PCIe x 4 que provee un bus eficiente para operaciones determinísticas y que, para aplicaciones que utilicen analizadores de red o nodos de procesamiento múltiple, se utilice la opción de la interfaz 10 GigE para un mejor desempeño. La arquitectura de la USRP X310 se muestra en la figura 4.5.

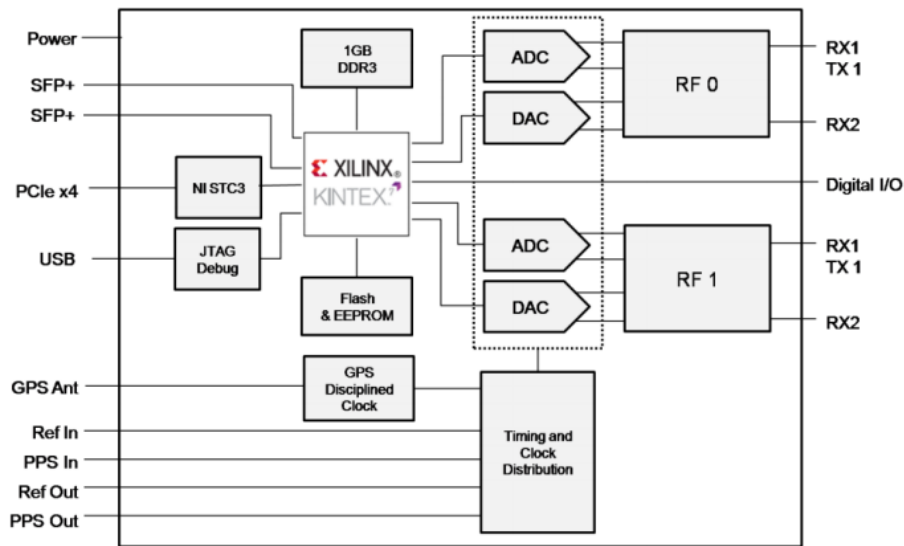


Fig 4.3 Arquitectura de la USRP X310 [38].

La X310 incluye además varias características adicionales que facilitan el desarrollo de sistemas inalámbricos. El DDR3 integrado de 1GB con acceso flexible a través del diseño de referencia de la FPGA, complementa los recursos de ésta, ofreciendo memoria intermedia y memoria para almacenar datos. Un GPSDO opcional interno proporciona una referencia de frecuencia de alta precisión y una alineación de sincronizaciones de hasta 50 ns cuando se sincroniza con el sistema GPS. El conector GPIO externo permite a los usuarios controlar componentes externos como amplificadores y conmutadores, aceptar entradas como desencadenadores de eventos y observar señales de depuración. Las USRP X310 también incluye un adaptador JTAG interno que permite a los desarrolladores de FPGAs cargar y depurar fácilmente nuevas imágenes a la FPGA del dispositivo.

Tabla 4.2 Comparación entre las FPGA del USRP N210 y USRP X310.

	USRP N210	USRP X310
FPGA	Spartan XC3SD3400A	Kintex 7 – 410T
Celdas lógicas	53k	406k
Memoria	2,268 Kb	28,620 Kb
Multiplicadores	126	1540
Razón de reloj	100 MHz	200 MHz
Ancho de banda de transferencia por canal (16 bit)	25 MS/s	200 MS/s

4.3 Tarjeta hija

Como se mencionó anteriormente, la tarjeta hija en el USRP realiza las funciones del procesamiento en radiofrecuencia. Un dispositivo USRP puede ser compatible con varias tarjetas hijas las cuales se diferencian principalmente en cuanto a frecuencia de trabajo, ganancia y ancho de banda.

4.3.1 SBX

La tarjeta hija utilizada con los dispositivos USRP N210 para el presente proyecto es la SBX, presentada en la figura 4.6, la cual es un dispositivo transceptor de un amplio ancho de banda que provee un máximo de 100 mW de salida de potencia y tiene una figura de ruido típica de 5 dB. Los osciladores locales para las cadenas transmisoras y receptoras operan independientemente, lo cual permite una operación full-dúplex en frecuencias de transmisión y recepción diferente. La SBX tiene capacidad MIMO y provee un ancho de banda de 40 MHz. Esta tarjeta es ideal para aplicaciones que requieran acceso a bandas de frecuencias que abarquen el rango desde los 400 MHz hasta los 4400 MHz. Algunos de los ejemplos de áreas de aplicación incluyen la tecnología WiFi, WiMAX y transceptores en la banda de frecuencias S y de 2.4 GHz [39].



Fig 4.4 Tarjeta hija SBX 400-4400 MHz Rx/Tx (40 MHz) [39].

El puerto de la antena transmisora se denomina TX/RX y para la antena receptora puede ser el TX/RX o RX2, sin embargo, si se utiliza la tarjeta en modo full-dúplex, la antena receptora deberá ser establecida siempre a RX2 independientemente de la configuración. La ganancia de transmisión y recepción abarcan ambas el rango de 0 a 31.5 dB.

4.3.2 CBX-120

La tarjeta hija CBX-120 (figura 4.7) será la ocupada con el USRP X310 en el presente proyecto y es un transceptor de banda ancha full-dúplex que cubre una banda de frecuencias de 1.2 GHz a 6 GHz con un ancho de banda de instantáneo de 120 MHz. La CBX se puede utilizar en una amplia gama de aplicaciones que incluyen las investigaciones en tecnologías WiFi, estaciones de bases celulares, radios cognitivas y RADAR.

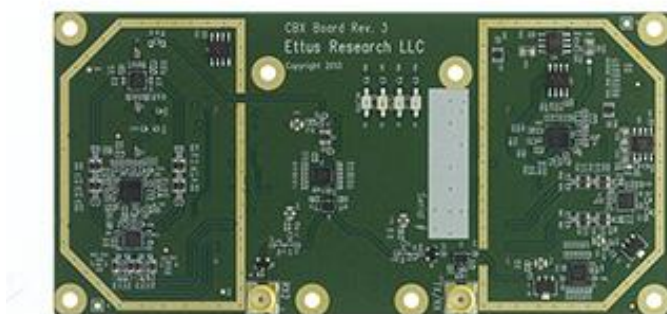


Fig 4.5 CBX 1200-6000 MHz Rx/Tx (120 MHz), solo serie X [40].

Los osciladores locales para las cadenas de recepción y transmisión funcionan independientemente, permitiendo la operación en múltiples frecuencias. Las referencias para los osciladores locales se derivan del reloj maestro USRP, lo que permite una operación coherente con múltiples dispositivos sincronizados USRP X300 / X310. El CBX-120 es compatible con los productos USRP X300 y USRP X310 de alto ancho de banda.



CAPÍTULO 5. GNU RADIO

GNU Radio es una herramienta de software de código abierto que contiene librerías de bloques para procesamiento de señales, los cuales son interconectados tanto para la construcción y despliegue de SDRs como para realizar simulaciones del procesamiento de señales. Los bloques de procesamiento de señales son escritos en C++ mientras que Python se utiliza como lenguaje de *script* para interconectar los bloques entre sí y formar el gráfico de flujo.

El presente capítulo presenta una descripción del software GNU Radio, el cual es el seleccionado para realizar la implementación de la capa física del estándar IEEE 802.11p con tecnología MIMO. Se definen las características principales del software, las herramientas más comunes y los *drivers* utilizados para comunicarse con la plataforma de hardware USRP.

5.1 Introducción

GNU Radio cuenta con una interfaz gráfica de usuario, similar a Simulink de Matlab, llamada GNU *Radio Companion* (GRC).

GNU Radio contiene varios bloques de procesamientos como filtros, codificadores de canal, elementos de sincronización, ecualizadores, demoduladores, decodificadores y demás elementos, los cuales son típicamente encontrados y ampliamente usados en sistemas de radio. Éste incluye un método para interconectar los bloques y gestionar cómo los datos son pasados desde un bloque hasta otro. Además, los usuarios de GNU Radio pueden crear y construir bloques para extender las funcionalidades generales del software. Los nuevos bloques pueden ser creados tanto en C++ o en Python. Cada bloque permite ser editado, actualizado o incluso implementado independientemente, sin interferir con la cadena general de comunicación.

La arquitectura base de GNU radio envuelve un gráfico de flujo complejo que consiste en módulos y algoritmos de bajo nivel. Dichos módulos utilizan la interfaz SWIG (*Simplified Wrapper and Interface Generator*) el cual es usado como intérprete entre el lenguaje C++ y Python como se muestra en la figura 5.1. La estructura de Python es responsable de la comunicación de datos a través de los módulos

de búfer y crea un planificador que ayuda a correr los bloques en un orden secuencial por iteraciones simples [41].

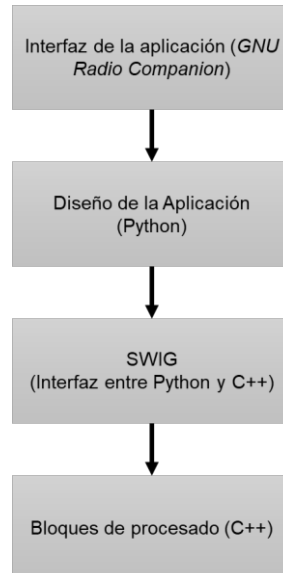


Fig 5.1 Arquitectura del Software de GNU Radio.

Las aplicaciones en GNU radio se pueden entender como un grafo donde los nodos simbolizan los bloques de procesamiento de señal. El software se conforma típicamente de cuatro elementos:

- **Fuente:** cada flujo de gráfico contiene al menos una fuente. La misma representa el comienzo de éste, y puede leer datos desde un USRP, un archivo de audio o puede ser implementado por un generador de señales, de mensajes, etc.
- **Sumidero:** cada flujo de gráfico debe tener al menos un sumidero. Éste constituye el destino final de la señal y puede ser un fichero, un despliegue gráfico o un sumidero USRP. Por ende, abarca los bloques de visualización de señales entre los que se encuentran sumideros para visualizar el espectro de las señales y las formas de ondas de las mismas.
- **Gráfico de flujos:** la aplicación se basa en un gráfico de flujos, el cual consiste en bloques intermedios entre la/las fuentes y el/los sumideros. Se pueden definir varios gráficos de flujos dentro de una sola aplicación.
- **Planificador (*scheduler*):** se crea por cada gráfico de flujo activo, el cual está basado en una cadena fija de flujo de datos entre los bloques. Es responsable de la transferencia de datos a través del flujo de gráfico. Monitorea cada bloque para tener datos suficientes en los buffers de entrada y salida tanto como desencadenar las funciones de procesamiento para esos bloques.

GNU Radio puede correr bajo varios sistemas operativos como Linux, Mac OSX y NetBSD.

5.2 GNU Radio Companion (GRC)

Como ya se ha comentado, GNU *Radio Companion* (GRC) es una interfaz de gráfico de usuario que se basa en la utilización de ventanas y bloques, y constituye específicamente la herramienta utilizada en la presente investigación. Los bloques procesan señales de forma continua, desde uno o varios puertos de entrada hasta uno o varios puertos de salida y se caracterizan por el número de puertos de entrada, el número de puertos de salida y el tipo de datos que fluye de un puerto a otro. Los tipos de datos más comunes que se manejan se muestran la figura 5.2, y los mismos se caracterizan por el color que presenta el puerto en la interfaz gráfica.

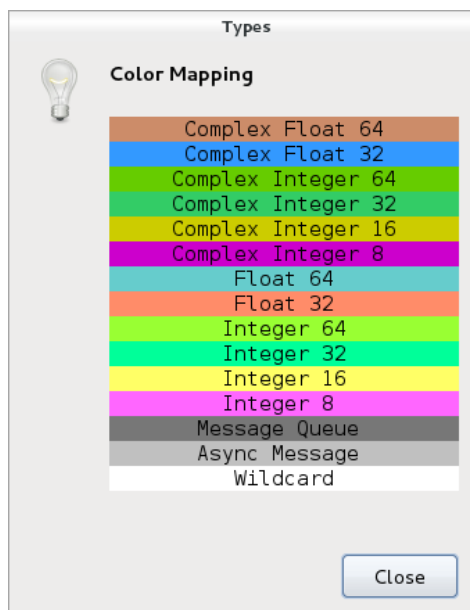


Fig 5.2 Código de colores de datos en GNU Radio.

GRC contiene una lista de módulos disponibles que se insertan en el panel de trabajo y pueden ser configurados de acuerdo a parámetros particulares de cada uno e incluso GRC señala si los mismos son incorrectos. Además, los módulos pueden ser interconectados muy fácilmente. Para acceder a la herramienta, se llama de manera simple desde un terminal escribiendo: “*gnuradio-companion*”.

En la ventana GRC ejecutada ya se permite realizar los diseños deseados y consta principalmente de cuatro zonas: la librería, la barra de Herramientas, el terminal y el espacio de trabajo como se muestra en la figura 5.3.

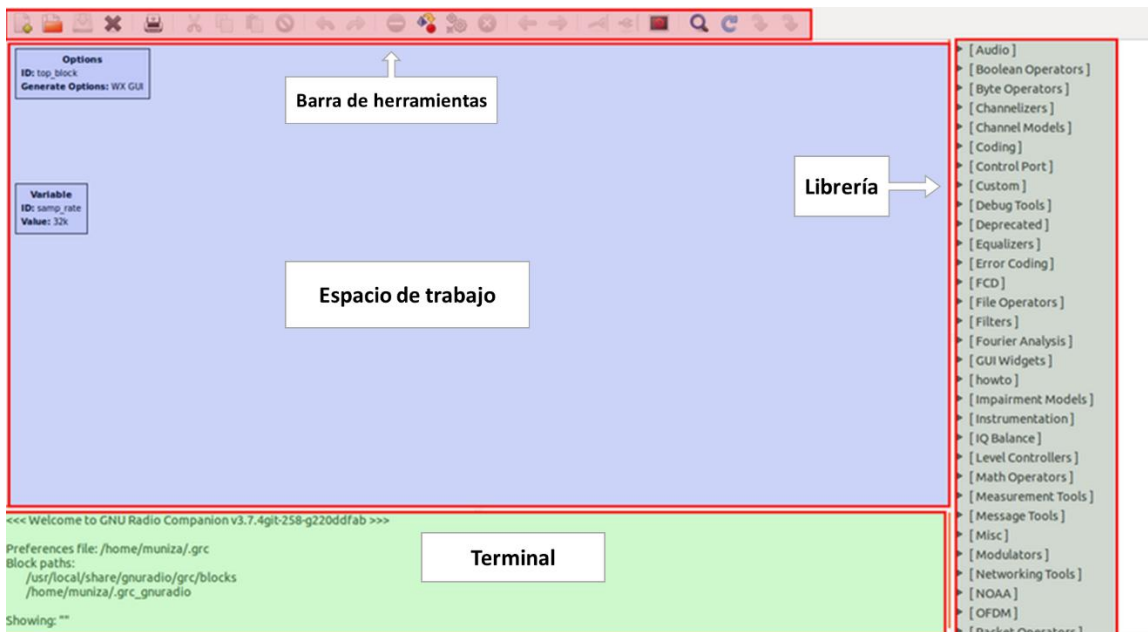


Fig 5.3 Estructura de la pantalla de inicio del GRC [42].

El bloque *Option* (izquierda superior del espacio de trabajo) permite la configuración de algunos parámetros generales de la implementación como título del proyecto y autor. El bloque *Variable*, es utilizado para establecer la razón de muestreo de la aplicación en general ya que cuando se construye un grafo las frecuencias de los diferentes bloques deben coincidir, de lo contrario se producen desfases en frecuencia de la señal generada.

Además de los bloques fuente y sumideros encontrados como elementos de GNU Radio y explicados de forma breve anteriormente, otros tipos de bloques presentes en GRC son:

- **Bloques de procesamiento de la señal:** realizan el procesamiento de la señal de entrada y la envían hacia la salida correspondiente. Pueden ser moduladores, demoduladores, filtros, correctores de errores o amplificadores, pueden realizar operaciones matemáticas, adicionar o eliminar etiquetas, etc.
- **Variables:** permiten variar los valores de ciertos parámetros etiquetados en la implementación con un nombre distintivo y único. Dicho valor puede ser escogido dentro de una lista de opciones mediante el bloque tipo *Chooser* o ser variable dentro de un rango acotado mediante la utilización de un bloque *Variable Slider*.
- **Agendas:** organiza los resultados de visualización de los diferentes bloques GUI sumideros en el grafo.

Después de ejecutar o generar la aplicación realizada, GRC genera el código de Python correspondiente a la misma en un archivo *.py cuyo código puede ser visto y modificado de forma directa.



A continuación, se explican de forma breve algunas de las librerías y módulos de GNU Radio utilizadas en la presente investigación.

5.2.1 Interfaz gráfica de usuario QT

El paquete *gr-qtgui* contiene varios bloques de interfaz gráfica basado en QT que constituyen sumideros gráficos de un gráfico de flujo de GNU Radio se utilizan de acuerdo a los propósitos de visualización como se muestra en la tabla 5.1 [43].

Tabla 5.1 Bloques de visualización QT en GNU Radio [43].

Tipo de Visualización	Clase de referencia en GNU radio	Características
Dominio del tiempo	<ul style="list-style-type: none"> ▪ gr::qtgui::time_sink_c ▪ gr::qtgui::time_sink_f 	En el eje-x se muestra el tiempo, en el eje-y la amplitud.
Dominio de Frecuencia o PSD	<ul style="list-style-type: none"> ▪ gr::qtgui::freq_sink_c ▪ gr::qtgui::freq_sink_f 	En el eje-x se muestra la frecuencia y en el eje-y la magnitud en dB.
Waterfall o espectograma	<ul style="list-style-type: none"> ▪ gr::qtgui::waterfall_sink_c ▪ gr::qtgui::waterfall_sink_f 	En el eje-x se muestra la frecuencia, en el eje-y el tiempo y en el eje-z la intensidad relacionada a la magnitud en dB.
Constelación	<ul style="list-style-type: none"> ▪ gr::qtgui::time_raster_sink_f ▪ gr::qtgui::time_raster_sink_b 	Tiempo vs. tiempo siendo el eje-z basado en el valor de la muestra
Histograma	<ul style="list-style-type: none"> ▪ gr::qtgui::histogram_sink_f 	Muestra un histograma del flujo de datos
Sumidero combinado	<ul style="list-style-type: none"> ▪ gr::qtgui::sink_c ▪ gr::qtgui::sink_f 	Combina la visualización del tiempo, frecuencia, <i>waterfall</i> y constelación.

5.2.2 Modulación Digital

Está contenida dentro del paquete *gr-digital*. Contiene todos los bloques de modulación digital, utilidades y ejemplos. Ello permite que GNU Radio soporte la creación y el uso de objetos de constelación para varias de las necesidades en comunicaciones digitales. Las constelaciones se definen con un grupo de puntos de constelación en el espacio complejo y el mapeo de símbolos a esos puntos [44].



Las clases de los objetos de las constelaciones se definen en *constelacion.h*, donde existe una jerarquía de las clases para diferentes propósitos y los cuales representan clases especiales de constelaciones. Cada clase de constelación tiene un conjunto de atributos y funciones útiles para manipular las constelaciones y para convertir los símbolos desde y hacia puntos complejos.

5.3 OFDM en GNU Radio

GNU Radio provee algunos bloques para transmitir y recibir señales moduladas OFDM. Los bloques están diseñados de una manera genérica, por ende, se permite lograr una funcionalidad específica mediante la parametrización correcta de los bloques disponibles, incluso, aunque en algunos casos se requieren de bloques más personalizados, la adición de funcionalidades propias es posible con muy pocas modificaciones [45].

5.3.1 Transmisión

Indexado de portadoras

Las portadoras comienzan siempre en el índice de la portadora de DC, la cual posee índice cero. Las portadoras a la derecha de DC son indexadas desde 1 hasta $N/2 - 1$. Las portadoras ubicadas a la izquierda, se indexan desde $-N/2$ hasta -1 . Utilizando dicha notación, permite que la longitud de la FFT sea variada sin realizar un cambio en la indexación de las portadoras.

Asignación de símbolos y portadoras

Varios bloques requieren el conocimiento de cuales portadoras serán asignadas, y si éstas transportarán datos o símbolos pilotos. GNU Radio utiliza tres objetos para ello en el bloque OFDM *Carrier Allocator*:

- occupied_carriers: es utilizada para establecer los símbolos de datos e identificar la posición de éstos dentro de un frame donde los datos y los símbolos pilotos son alternados.
- occupied_carriers[0] identifica cuales portadoras son ocupadas en el primer símbolo OFDM, occupied_carriers[1] realiza lo mismo en el segundo símbolo y así sucesivamente.
- pilot_carriers: se utilizan para establecer en cuales portadoras se encuentran los símbolos pilotos.
- pilot_symbols: se establece el valor que tomarán los símbolos en las portadoras pilotos

El bloque *OFDM Carrier Allocator* clasifica los escalares entrantes complejos en portadoras OFDM y también coloca los símbolos pilotos en las posiciones especificadas. Además, contiene la opción de pasar los símbolos OFDM que se añaden delante de cada trama, es decir, el preámbulo utilizado para la detección, sincronización y estimación del canal. A la salida entonces se tienen símbolos OFDM (vectores complejos de longitud FFT) que tienen que ser convertidos al dominio del tiempo antes de continuar, por lo que se envían a un bloque FFT.



Corrimiento FFT

En los casos donde los símbolos OFDM se pasan entre bloques, el comportamiento predeterminado es correr mediante la FFT estos símbolos, de manera que la portadora DC se encuentre en el centro ($N/2$, donde N es la longitud FFT) y el índice de portadoras comience en 0. Este corrimiento se realiza debido a que algunos bloques requieren reorganizar los símbolos de esta forma tanto por coherencia como por su funcionamiento. Por ende, este funcionamiento se utiliza en todos los bloques que pasan símbolos OFDMs.

Esta función se encuentra contenida dentro del paquete *gr-fft*. Contiene los bloques de procesamiento de las señales para realizar FFTs o funcionalidades relacionadas.

Adición del prefijo cíclico (CP)

Finalmente, los símbolos OFDMs deben ser adicionados con el prefijo cíclico.

5.3.2 Recepción

La recepción requiere de mayor procesamiento. En primer lugar, se realiza la FFT para transportar los símbolos OFDM al dominio de la frecuencia, donde se realiza el procesamiento de la señal, y luego se realiza el proceso de estimación del canal, corrimiento de la frecuencia y se eliminan los pilotos y el preámbulo, función que comprenden los bloques *OFDM Frame Equalizer* y *OFDM Serializer* de GNU Radio. Posteriormente los símbolos pueden ser convertidos a bits o ser enviados a un decodificador corrector de errores.

5.3.3 Transmisión de paquetes

En varios casos, la capa PHY de un transceptor utiliza el envío de paquetes en transmisión, en vez de datos enviados continuamente, por tanto, GNU Radio contiene este tipo de transmisión provista entre otros, por los bloques *Message Passing* y *Tagged Stream Block*. Con este tipo de herramientas, el esquema de transmisión de paquetes digitales resulta en una simple implementación permitiendo la creación de enlaces de comunicaciones basados en paquetes e incluso, de redes. Para entender a detalle el funcionamiento de los mismos se recomienda remitirse a [46].

5.4 Interfaz *USRP Hardware Driver* (UHD)

GNU Radio cuenta con el paquete UHD. Es la interfaz de la biblioteca UHD para conectarse, enviar y recibir datos entre la línea de productos de Ettus Research, LLC.

El objetivo del UHD es proporcionar un API (del inglés *Application Programming Interface*) y un driver para controlar el USRP. Este driver puede ser utilizado de manera *stand-alone* (no requiere un sistema



operativo para ejecutarse) a través del API o recurriendo a algún entorno o plataforma de desarrollo software como GNU Radio, LabVIEW, Matlab o Simulink.

El UHD acepta especificaciones de los subdispositivos, los cuales son cadenas usadas para configurar el dispositivo USRP de la forma deseada. En el contexto de UHD, se puede seleccionar un *front-end* mediante el uso de las especificaciones de un subdispositivo cuando se crea un instancia de un dispositivo USRP. Ello se logra con una lógica de enrutamiento dentro del FPGA que multiplexa los datos de acuerdo con la especificación del subdispositivo en cuanto a la ranura de tarjeta hija (en caso que aplique, ya que algunos USRP contienen dos ranuras, A y B, para conectar dos tarjetas hijas dentro de un mismo dispositivo) y el *front-end* de la tarjeta hija especificado por el designador de ranura para el mismo. Los *front-ends* de cada tarjeta hija se resumen en direcciones TX/RX, selección válida de antenas y conector RF asociado [47]. La intersección entre GNU Radio y el USRP lo constituyen un grupo de bloques en el paquete *gr-uhd*, el cual puede instalarse en Ubuntu mediante el comando:

```
“sudo apt-get install libuhd-dev libuhd003 uhd-host”
```

e incluye:

- *UHD: USRP Source block*: provee los datos recibidos para enviarlos a los bloques de procesamiento. Éste recibe las muestras y las pasa a un flujo/cadena de datos. Además, provee llamadas API para las opciones de recepción.
- *UHD: USRP sink block*: acepta los datos a transmitir desde bloques de procesamientos. Éste lee un flujo y los transmite en forma de muestras. También provee llamadas API para las opciones de transmisión.



CAPÍTULO 6. DESARROLLO

Este capítulo presenta una descripción detallada de la implementación del sistema MIMO 2x1 en una plataforma SDR usando el software GNU Radio. Se definen la estructura del sistema propuesto, así como las funciones realizadas por cada componente de la capa PHY IEEE 802.11 propuesta.

6.1 Sistema SISO 802.11p

Como punto de partida en la presente investigación, se toma en cuenta el trabajo realizado en [10] y que posteriormente fue extendido en [11], donde se implementa la capa física de un transceptor 802.11p en SDR, con código abierto utilizando la herramienta GNU Radio. La contribución del mismo se puede resumir en que constituyó el primer transceptor basado en SDR para el estándar 802.11p, utilizando computadoras simples para realizar el procesamiento y logrando modulaciones hasta el orden de 64 QAM $\frac{3}{4}$, con distintos tamaños de paquetes y con un ancho de banda de 10 MHz, que corresponde al estándar, aunque puede llegar hasta los 20 MHz.

6.1.1 Transmisor

En el trabajo realizado en [11], el autor presenta un bloque jerárquico en transmisión el cual representa la capa física PHY para permitir una estructura más clara en GNU *Radio Companion*. Para este proyecto, se decidió realizar una reestructuración del mismo, de manera que todos los bloques quedaran dentro de un mismo gráfico de flujo, con el objetivo de entender mejor la estructura general del sistema al añadir los bloques STBC para la codificación MIMO tanto en transmisión como en recepción. El esquema del transmisor se muestra en la figura 6.1.

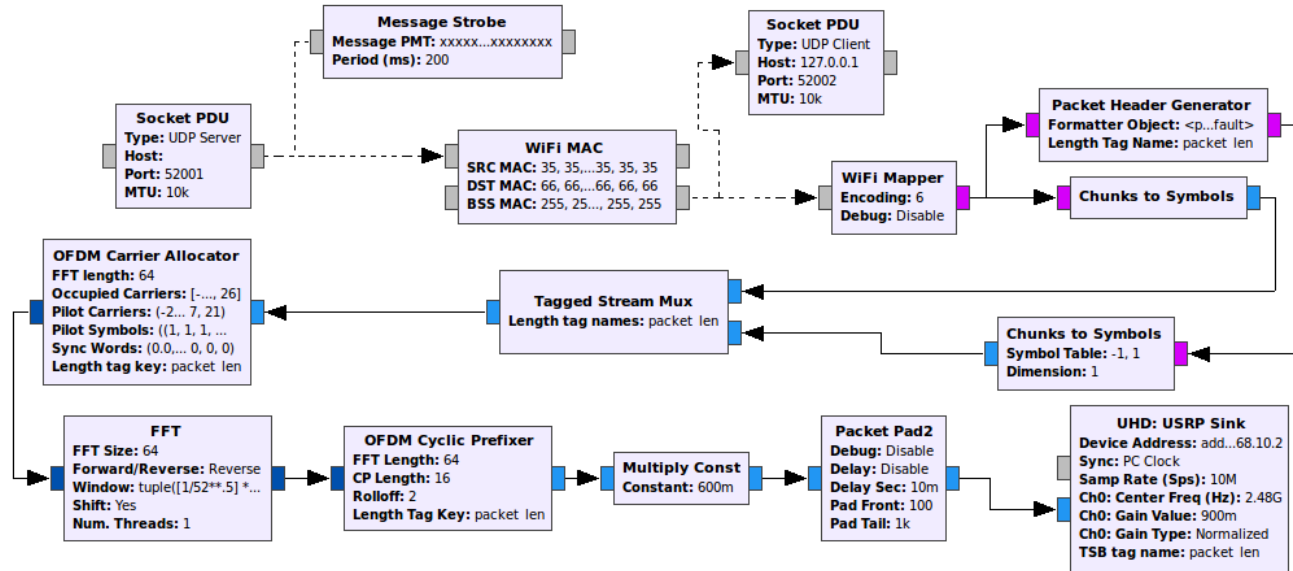


Fig 6.1 Transmisor implementado en GNU Radio del sistema SISO.

Al comienzo de la cadena de transmisión se encuentran el bloque *Socket PDU* y *Message Strobe*. El primero se utiliza para establecer un puerto UDP de destino para los paquetes enviados, mientras que en el último se establece el período y la longitud del mensaje a transmitir. Posteriormente se le adicionan las direcciones que corresponden a los campos *Address* dentro de la trama MAC, cuyo objetivo es lograr una interoperabilidad con tarjetas 802.11 comerciales.

El bloque *Wifi Mapper* es el responsable de la generación de los campos *RATE* y *SERVICE* ubicados en el encabezado PLCP, explicados en la sección 3.4.2, así como del entrelazado (*interleaving*), la codificación convolucional y la aleatorización (*scrambling*) de los datos. Lo que significa que es el encargado de recibir el esquema seleccionado de codificación y modulación como entrada. Posteriormente se encuentra el bloque *Packet Header Generator*, el cual genera el encabezado del frame incluyendo los campos *SIGNAL* y *SERVICE*. El encabezado es modulado en BPSK por el bloque *Chunks to Symbols* superior y el frame restante es modulado de acuerdo a la modulación seleccionada por el bloque *Chunks to Symbols* inferior. Luego, el encabezado es finalmente unido al frame restante.

El siguiente bloque es el *OFDM Carrier Allocator* que es responsable de la agregación de los pilotos de las subportadoras y las secuencias de entrenamientos. Las asignaciones de las mismas se establecieron según lo explicado en la sección 3.4 de este trabajo, siguiendo las especificaciones del estándar.

A continuación en el diagrama de flujo, se encuentra el bloque *FFT* que realiza la *FFT* Inversa para transformar del dominio de la frecuencia al dominio del tiempo, el bloque *OFDM Cyclic Prefixer* agrega el intervalo de guarda a cada símbolo del frame y, finalmente, el bloque *USRP Sink* define los

parámetros en la tarjeta SDR USRP, tales como la antena del transmisor, el reloj SDR y la razón de muestreo.

6.1.2 Receptor

La estructura del receptor se muestra en la figura 6.2. El receptor se divide en dos partes: la parte superior que es responsable de la detección del frame y la parte inferior que es responsable de la decodificación del frame.

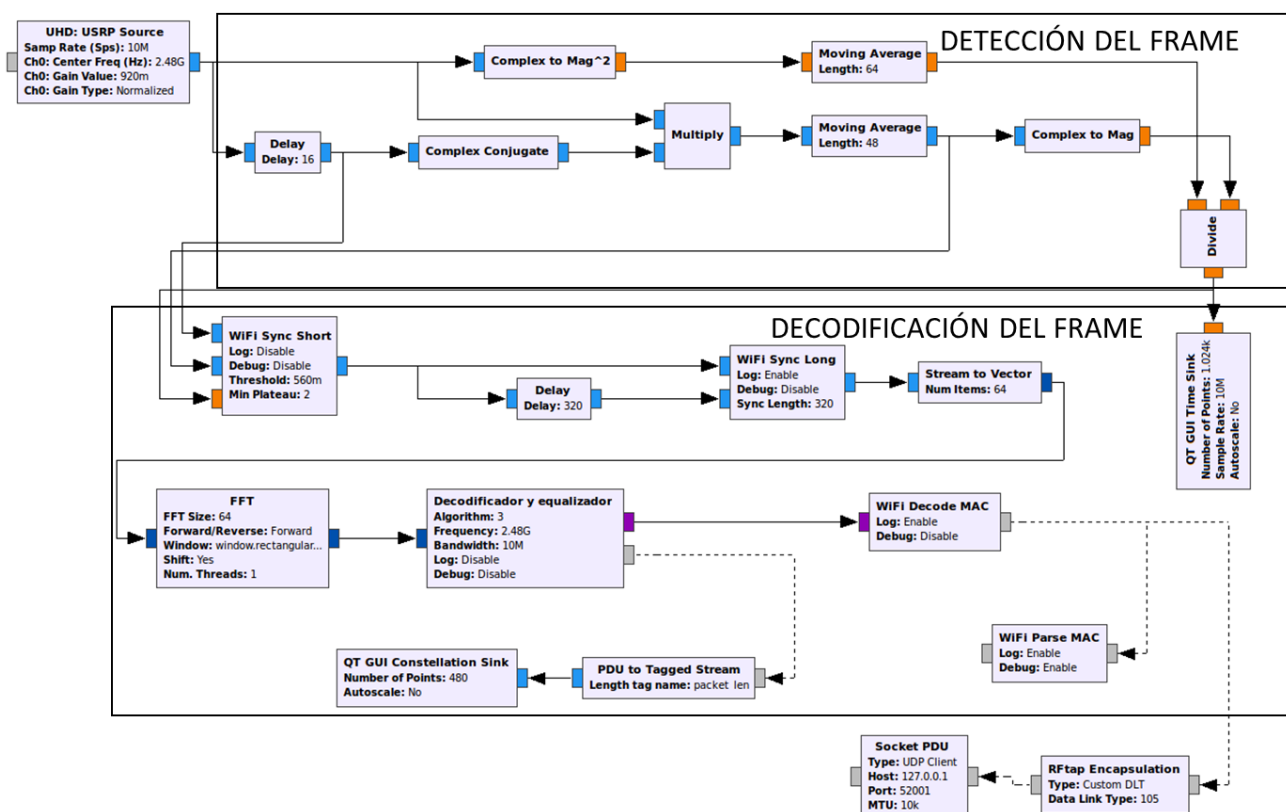


Fig 6.2 Esquema del receptor en GNU Radio del sistema SISO.

Para entender cómo se realizan esos pasos es importante conocer la composición del frame IEEE 802.11 y en particular el preámbulo del protocolo de convergencia de la capa física OFDM. Es importante señalar que el receptor a diferencia del transmisor, requiere de alta capacidad computacional por lo que se recomienda utilizar VOLK (*Vector Optimized Library of Kernels*), presentado en [48] como una biblioteca matemática de fácil uso para “una instrucción, múltiples datos” (SIMD, por sus siglas en inglés) y como una estructura para el desarrollo de código abierto de código SIMD. VOLK proporciona una capa de abstracción para las operaciones SIMD específicas del hardware. Es decir, en vez de realizar cálculos iterativos, las instrucciones SIMD actúan sobre vectores de datos proveyendo una aceleración significativa en el procesamiento de los mismos [11].



La primera tarea del receptor es detectar el comienzo de un frame OFDM basado en el algoritmo de detección del frame que ha sido introducido en el algoritmo 1 mostrado en [49]. Este algoritmo se basa en la autocorrelación de la secuencia de entrenamiento corta. Cada trama del frame IEEE 802.11 a/g/p comienza con una secuencia de preámbulo corto, la cual consiste en un patrón que abarca 16 muestras y las repite 10 veces. El receptor hace uso de este patrón cíclico y calcula el valor de la autocorrelación 'a' de la cadena entrante muestreada 's' en el intervalo 16 mediante la sumatoria de los coeficientes de autocorrelación sobre una ventana N_{win} ajustable, (donde, \bar{s} denota el complejo conjugado de s) [10].

$$a[n] = \sum_{k=0}^{N_{win}-1} s[n+k] \bar{s}[n+k+16] \quad (6.1)$$

De manera que no afecte el nivel absoluto de las muestras entrantes, se realiza una normalización de la autocorrelación con una potencia promedio. En el receptor se considera que existe un nivel alto de señal si tres muestras consecutivas se encuentran sobre un umbral configurable. Si se detecta un frame, un número fijo de muestras se conducen al bloque subsecuente.

Los ocho bloques involucrados en la detección del frame diseñados en [10] hacen uso de la biblioteca VOLK, lo cual permite que la implementación soporte razón de muestreo de 20 Msps para IEEE 802.11a/g y 10 Msps para IEEE 802.11p. De esta forma, la velocidad de procesamiento incrementada con el uso de esta biblioteca es crucial para el receptor, ya que todos los bloques involucrados tienen que procesar el flujo de muestreo proveniente del USRP a máxima velocidad.

Posterior al cálculo del coeficiente de autocorrelación, el bloque siguiente es el *WiFi Sync Short*, cuya entrada son las muestras del USRP y el coeficiente de autocorrelación normalizado. Si se detecta una coincidencia (es decir, el flujo de la autocorrelación supera el umbral establecido), el bloque conduce un número fijo de muestras hacia el bloque subsecuente del gráfico de flujo, de lo contrario, las muestras se descartan.

El bloque siguiente en la implementación del receptor es el *WiFi Sync Long*, el cual es responsable de la corrección *offset* en frecuencia y la alineación de símbolos. La corrección *offset* en frecuencia es necesaria, ya que el oscilador local del transmisor y el receptor deben trabajar en frecuencias ligeramente diferentes. Para compensar eso, el autor utiliza el algoritmo introducido en [50], el cual estima la corrección *offset* en frecuencia basado en la secuencia de entrenamiento corta. Idealmente, durante la secuencia corta, una muestra $s(n)$ debería corresponder con la muestra $s(n+16)$ debido a su propiedad cíclica. Sin embargo, eso no ocurriría si se introduce ruido y corrimiento en frecuencia, por tanto, para estimar el valor final, se aplica el promedio y el valor final para la frecuencia *offset* df se calcula (siendo N_{short} la longitud de la secuencia de entrenamiento corta) [50]:

$$df = \frac{1}{16} \arg(\sum_{n=0}^{N_{short}-1-16} s[n] \bar{s}[n+16]) \quad (6.2)$$



Finalmente, se aplica la compensación en frecuencia a cada muestra como:

$$s[n] \leftarrow s[n]e^{i(ndf)} \quad (6.3)$$

Alineación de símbolos

El bloque *WiFi Sync Long* es también responsable de la alineación de símbolos. Cada símbolo OFDM contiene 80 muestras, 16 muestras del prefijo cíclico y 64 de las muestras de datos. El proceso de alineación de símbolos consiste en el cálculo del comienzo de un símbolo, la extracción de los símbolos de datos e insertarlos a un algoritmo haciendo la transformada rápida de Fourier (FFT). Además, la alineación se realiza con la ayuda de la secuencia de entrenamiento larga, la cual se compone de un patrón largo de 64 muestras que se repite cada 2.5 veces.

Finalmente, conociendo el comienzo del primer símbolo de datos, el bloque puede extraer los símbolos de datos y la parte de las muestras de datos que corresponden a un símbolo se pasan al bloque siguiente en el gráfico de flujo, el cual reconoce el comienzo del frame ya que se etiqueta el primer símbolo de cada frame OFDM. De esta forma, se extrae el prefijo cíclico al agrupar solamente las muestras que correspondan a los símbolos de datos individuales (es decir, que no incluyan los primeros 16 símbolos) [10].

Corrección de corrimiento de fase

El siguiente bloque en la cadena receptora es el bloque FFT, el cual es responsable de la transición del dominio del tiempo al dominio de la frecuencia. El primer bloque en el dominio de la frecuencia es el *WiFi Frame Equalizer*, el cual aplica la corrección del corrimiento (*offset*) en fase y la estimación del canal. Como los tiempos de muestreo del transmisor y el receptor no están sincronizados y como la alineación de símbolos no es perfecta, se introduce un corrimiento de fase, el cual es lineal con la frecuencia y puede ser corregido con la ayuda de subportadoras pilotos. IEEE 802.11 impone 4 subportadoras pilotos que codifican una constelación BPSK predefinida, la cual es la misma para cada frame, pero es diferente de símbolo a símbolo. Por ende, se requiere saber el índice de símbolo del frame, el cual será asignado por una etiqueta en el flujo de muestras que es adicionado por el bloque *WiFi Sync Long*. El *offset* de fase se estima por una regresión lineal y es compensado basado en las 4 subportadoras pilotos [10].

Estimación de canal

El bloque *WiFi Frame Equalizer* es también responsable de la estimación del canal, ya que la magnitud de la portadora tiene que ser corregida. Esta estimación es especialmente importante si se usan altas codificaciones tal como 16-QAM y 64-QAM, donde la magnitud transporta información.

El bloque *WiFi Frame Equalizer* también elimina DC, subportadoras piloto y de guarda y agrupa el vector de entrada de 64 símbolos en 48 símbolos.



Decodificación del campo *SIGNAL*

El bloque siguiente en la cadena receptora es el *OFDM Decode Signal*. En cada frame, después de las secuencias de entrenamiento cortas y largas, existe un campo *SIGNAL*, el cual es modulado en BPSK y codificado en un símbolo OFDM con una razón de $\frac{1}{2}$ que contiene información sobre la longitud y la codificación de los símbolos siguientes. Si el campo *SIGNAL* se decodifica exitosamente, este bloque acota el flujo de muestras con una etiqueta, transportando una tupla de codificación y longitud del frame. Dicha etiqueta es usada por el bloque siguiente para decodificar la carga útil.

Decodificación de la trama

La tarea final en la cadena receptora es la decodificación de la carga útil. Esta tarea es realizada por múltiples bloques. El bloque *WiFi Decode MAC* recibe vectores de 48 puntos de constelación en el plano complejo, correspondiente a las 48 subportadoras de datos por símbolo OFDM. Dependiendo del esquema de modulación y codificación (MCS), se mapean los bits de un símbolo (demodulación) y luego se permutan para llevar a cabo el desentrelazado de los mismos como operación inversa al entrelazado (*Interleaving*) realizado durante la transmisión.

Luego se realiza la decodificación de la señal y se empaqueta la carga útil en un mensaje GNU Radio y se envía a los siguientes bloques en el gráfico encargados de monitorear los frames en tiempo real, de manera que se pueda observar información detallada sobre los mismos.

El bloque *Tuntap PDU* es responsable del envío de mensajes recibidos por GNU Radio desde el SDR a la interfaz virtual a la cual es asignada. Además, al configurar una dirección IP a la interfaz virtual permite al usuario enviar paquetes desde la parte transmisora, debido a que los mismos serán reenviados a GNU Radio, encapsulado con parámetros que han sido establecidos y por ende enviados al SDR y transmitidos por el medio inalámbrico.

6.2 Sistema MIMO propuesto

Aunque el estándar 802.11p no comprende un entorno multiantena, la técnica MIMO podría ser prometedora para la redes vehiculares, de manera que pueda brindar solución a varios de los desafíos presentes en este tipo de redes [13].

Como se explica anteriormente, para crear el sistema MIMO en la presente investigación, se utilizó codificación espacio-temporal ocupando el esquema propuesto por Alamouti. Ello permite que el sistema implementado contenga una ganancia de diversidad espacial que constituye la habilidad de mitigar el desvanecimiento por multitrayecto mediante el uso de múltiples antenas para mejorar la calidad y la confiabilidad de un enlace inalámbrico. Esta ganancia aumenta debido a los múltiples canales que pueden ser observados en un receptor de la misma señal transmitida.

A continuación, se muestra un diagrama general del sistema MIMO 2x1 propuesto:

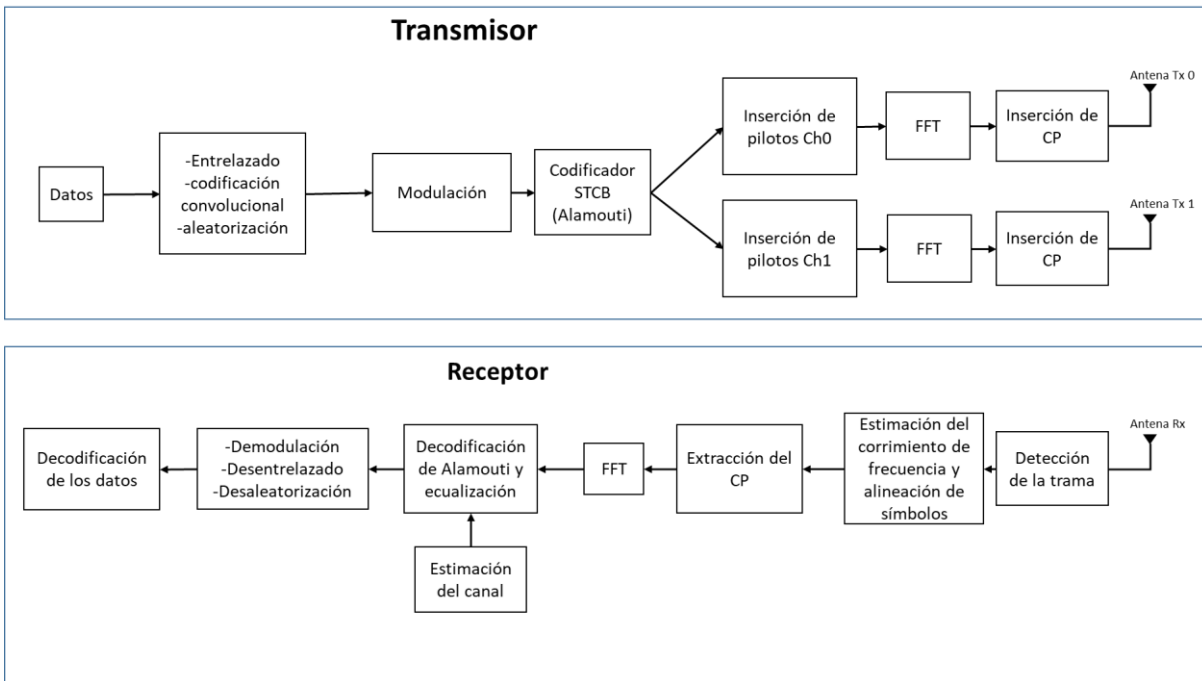


Fig 6.3 Esquema del sistema MIMO propuesto.

6.3 Implementación en software

La implementación utiliza los mismos bloques implementados en el sistema SISO OFDM [11] pero con la adición de los bloques de codificación/decodificación Alamouti, además de algunos cambios que serán explicados a continuación.

6.3.1 Transmisor

Para realizar la codificación de Alamouti en el transmisor se implementaron dos bloques "out-of-tree" en GNU Radio denominados "Codificación Canal 0" y "Codificación Canal 1". En los mismos se realiza la codificación propuesta por Alamouti explicada en la sección 3.5.2, dando a la salida los datos que son transmitidos por la antena 0 y la antena 1 respectivamente. La implementación de los bloques se realizó utilizando lenguaje C++, ya que tiene más sentido tomando en cuenta que es importante el rendimiento del sistema [51]. A continuación, se muestra la implementación del transmisor realizada en GNU Radio.

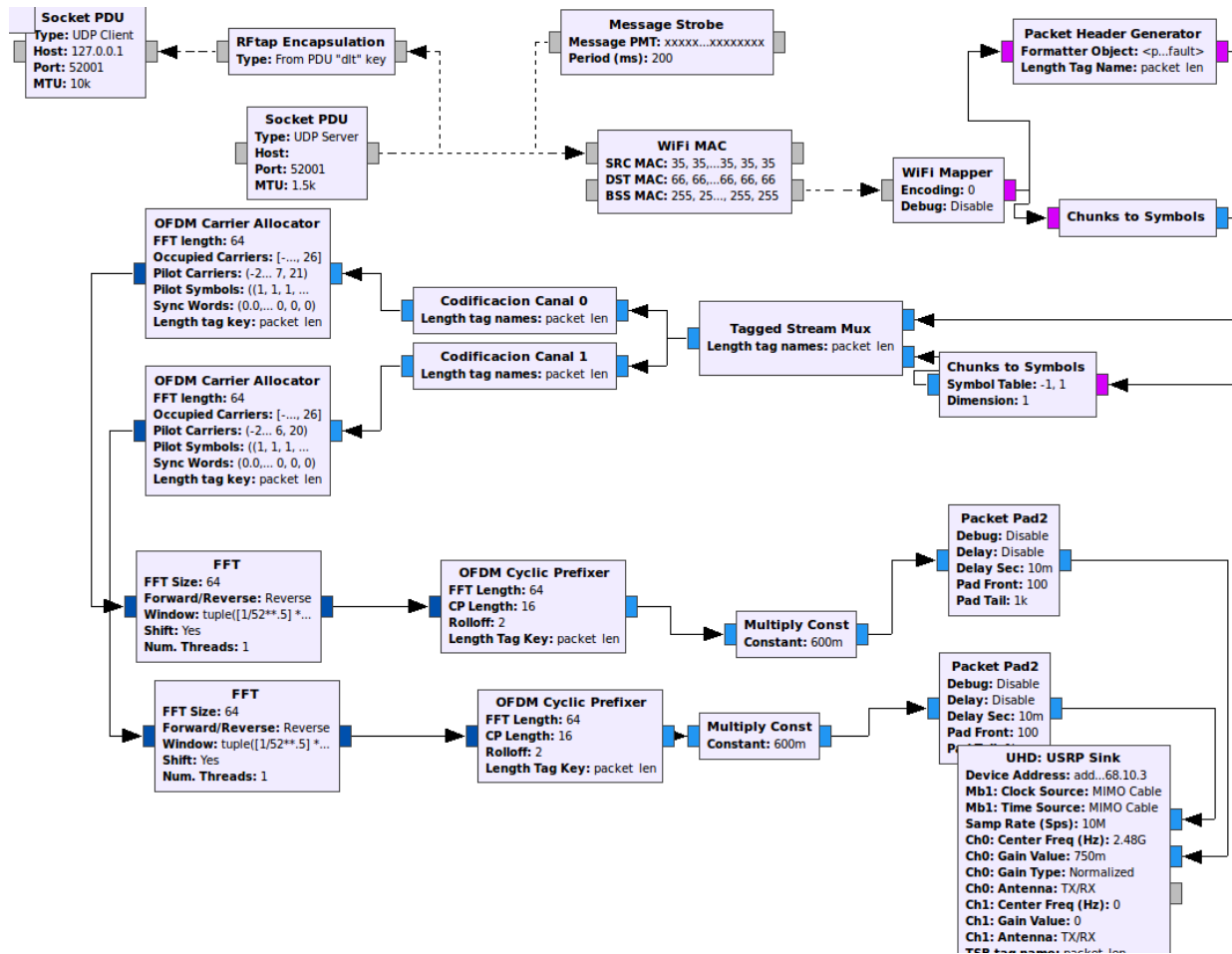


Fig 6.4 Implementación en GNU Radio del transmisor propuesto.

Cómo se observa en la figura 6.4, los paquetes transmitidos fueron monitoreados en el puerto 52002 de la interfaz *loopback* de la PC con ip 127.0.0.1. Para el presente trabajo se ocupa el software analizador de redes Wireshark.

Los símbolos OFDM transmitidos transportan cada uno los símbolos pilotos, los cuales se insertaron en posiciones no coincidentes, pero localizados de manera continua respecto a la otra antena y ubicando un cero en el índice en que la otra antena envía una señal piloto. De esta forma, cuando una antena está transmitiendo una señal piloto la otra se encuentra transmitiendo un nulo, lo cual permite estimar los dos canales en el receptor utilizando un conjunto diferente de señales pilotos para cada canal. Dicho patrón de inserción de pilotos de forma combinada se muestra de manera general en la figura 6.5 y se define en [52].

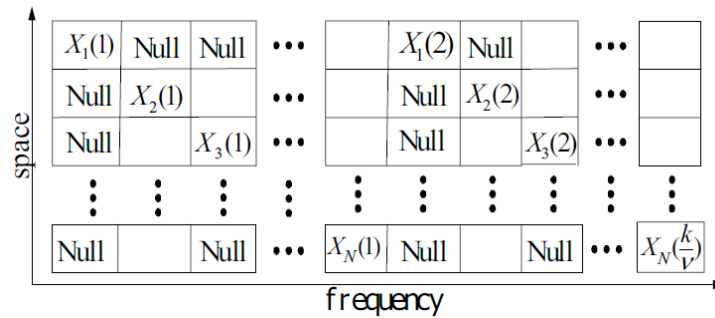


Fig 6.5 Patrón ortogonal de pilotos.

Dicho esquema muestra que los pilotos y sus posiciones en diferentes antenas son ortogonales a cada una, donde $X_i(v)$ corresponde a la posición del piloto siendo v el número de piloto e i la antena transmisora. Por ende, las señales en las posiciones pilotos no pueden ser afectadas por las señales de las demás antenas transmisoras para las antenas transmisoras y receptoras dadas y la estimación del canal de un sistema MIMO puede ser transferida a un estimador de canal absoluto de un sistema SISO.

En el presente trabajo, las subportadoras pilotos se ubican en las posiciones -21, -7, 7 y 21 para la primera antena, y en las posiciones -20, -6, 6 y 20 para la segunda antena, por ende, esta última contendrá nulos en las posiciones pilotos de la primera antena y de igual forma, la primera antena contendrá nulos en las posiciones de los pilotos de la segunda antena. Tal distribución será utilizada en la etapa de recepción para estimar los dos canales correspondientes a las antenas transmisoras de forma independiente, tomando en cuenta la distribución de pilotos de la antena 0 para estimar el canal 0 (desde antena transmisora 0 a la antena receptora), y la distribución de pilotos de la antena 1 para estimar el canal 1 (desde antena transmisora 1 a la antena receptora).

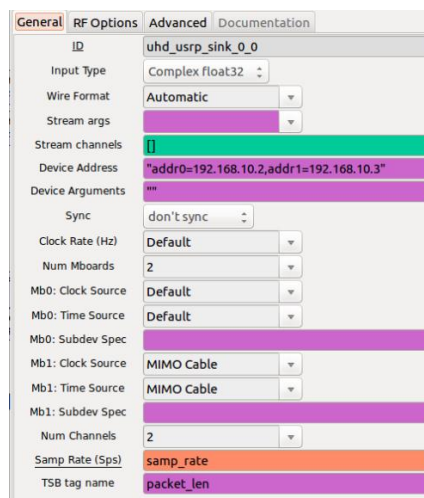


Fig 6.6 Configuración MIMO del bloque UHD: USRP Sink.



En la figura 6.6 se muestra la configuración establecida para el bloque *UHD: USRP Sink* en GNU Radio.

Para la transmisión se utilizaron los dos dispositivos USRP N210 en configuración MIMO como se ha mencionado anteriormente, los que de manera predeterminada traen configurado la dirección IP privada 192.168.10.2; por tanto, para que ambos dispositivos se comunicaron con la PC, fue necesario cambiarle la dirección IP a uno de los dos a 192.168.10.3. Dicho parámetro fue especificado en el bloque para que se estableciera la comunicación con los mismos de manera conjunta. Se definieron además el número de *motherboards*, siendo dos en este caso para transmisión y como fuente de sincronismo se define el cable MIMO (figura 6.9), encargado de establecer automáticamente el reloj de referencia de un equipo hacia el otro.

6.3.2 Receptor

En la etapa de recepción, se mantienen las mismas etapas de sincronización, estimación CFO, eliminación del prefijo cíclico y la transformada FFT que en el sistema SISO. La tarea principal fue realizar la decodificación de Alamouti y obtener de manera más exacta los coeficientes de estimación del canal, ya que de ello depende la correcta decodificación de los datos.

El bloque *Wifi Frame Equalizer* fue modificado para realizar la decodificación de Alamouti y para ello se utilizó el segundo estimador expuesto en la sección 3.6.1 que utiliza los pilotos ubicados entre las portadoras de datos. Como se explica anteriormente, la estimación para cada uno de los canales se realiza en función de la ubicación de los portadores pilotos establecidos para cada antena. También otra modificación fue reagrupar el vector de 64 símbolos en 44 símbolos (en vez de 48) a la salida del bloque, debido a la utilización de 4 subportadoras nulas adicionales para el sistema MIMO. La implementación en GNU Radio se muestra en la figura 6.7, y se denominó al bloque modificado cómo “Decodificador y ecualizador”.

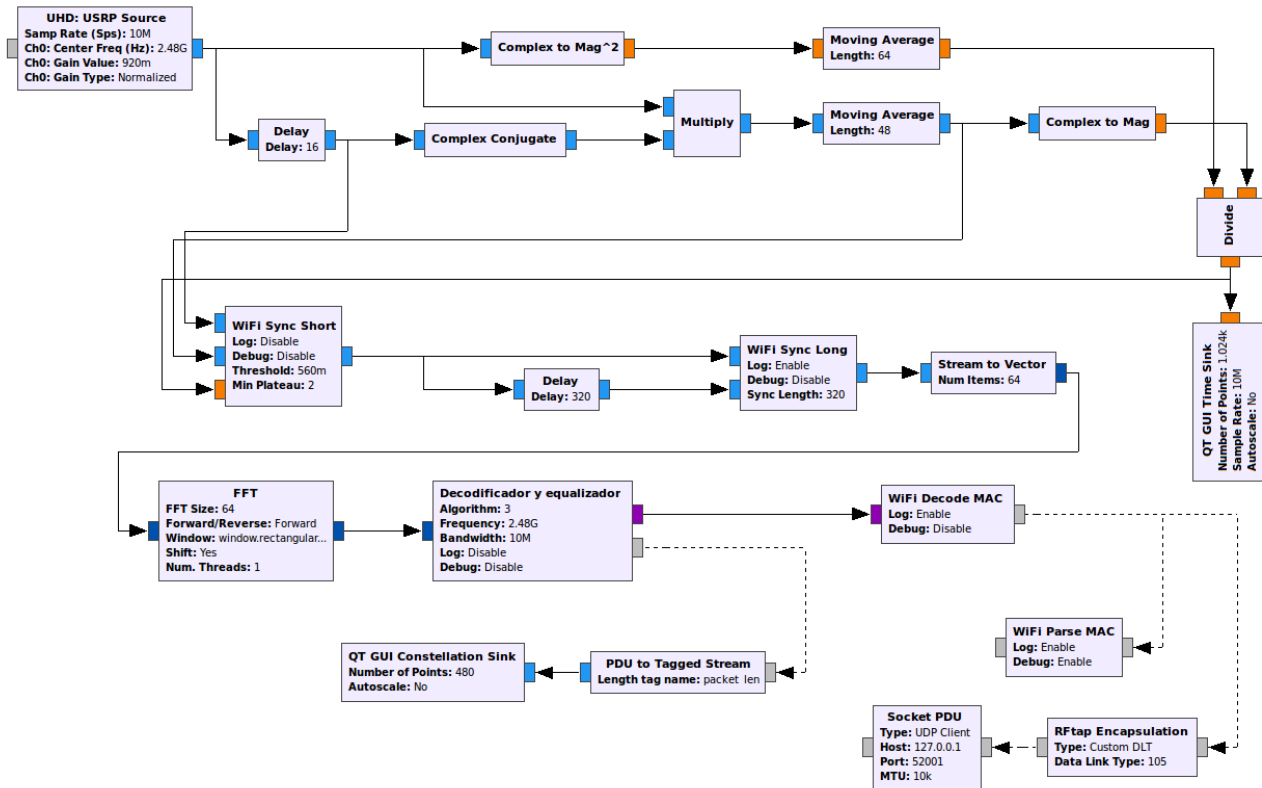


Fig 6.7 Implementación en GNU Radio del receptor propuesto.

6.4 Implementación en Hardware

La implementación en hardware se realiza utilizando dos equipos de Ettus USRP N210 para la transmisión y un USRP X310 para la recepción. Para ello se tomó en cuenta las indicaciones de la página de Ettus Research, de que los equipos USRP N210 son los más adecuados para trabajar en configuración MIMO y además, el USRP X310 tiene mayor capacidad de procesamiento, por ende, es idóneo para trabajar en la recepción por la cantidad de datos que necesitan ser procesados en tiempo real.

A continuación, se muestra el diagrama general de la implementación realizada.

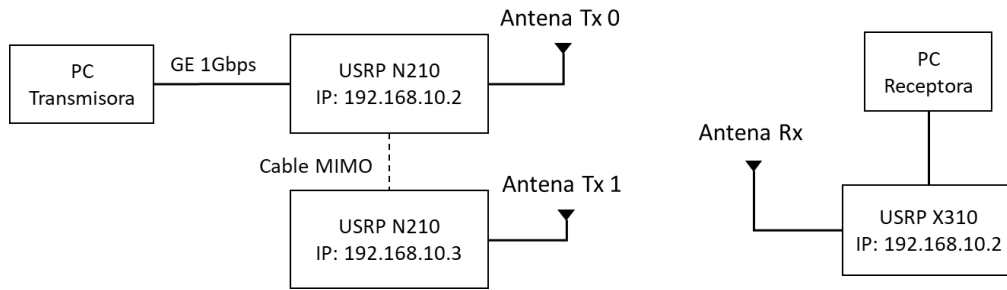


Fig 6.8 Diagrama general de la implementación MISO.

El cable de expansión MIMO (figura 6.9) se utiliza para unir un par de USRP N2xx, solo se necesita un cable por cada par de dispositivos enlazados (no es compatible con otra serie de USRP).



Fig 6.9 Cable de expansión MIMO utilizado en la implementación.



CAPÍTULO 7. RESULTADOS

En el presente capítulo se presentan las pruebas realizadas para evaluar el desempeño del sistema MIMO propuesto en este trabajo de tesis. Para ello se explica detalladamente la forma en la que se recogieron los resultados y las herramientas utilizadas.

7.1 Recolección de resultados

Para documentar y analizar los resultados se utilizó el protocolo RFtap, el mismo está diseñado para proveer metadatos de RF sobre los paquetes recibidos, tales como [53]:

- Potencia de señal y ruido
- Información de fase y sincronización
- Frecuencias *Doppler* y de portadora de cada paquete.

El protocolo salva la brecha existente entre plataformas de Radios Definidos por Software y las herramientas convencionales de monitoreo de redes y paquetes. En el presente trabajo se utilizó para analizar los datos provenientes de GNU Radio en Wireshark. Para ello se realizó la correspondiente instalación del módulo en GNU Radio, el cual implementa el bloque *RFtap Encapsulation*, utilizado para encapsular los metadatos de RF sobre los paquetes para Wireshark. El bloque se puede conectar a cualquier puerto de salida del tipo *PDU Message*, en este caso será conectado al bloque *Socket PDU* de GNU Radio, que permite establecer un host y puerto de destino para los datos que se reciben. En la figura 7.1 se muestran las opciones disponibles para configurar el bloque.



Fig 7.1 Bloque de encapsulación RFTap.

Para interpretar correctamente los paquetes de datos, Wireshark necesita saber el tipo de paquete que se analizará. Para el presente caso, se establece el identificador 105 (DLT_IEEE802_11) perteneciente al *Data Link Type* (DLT) para el estándar IEEE 802.11 de LAN inalámbrica establecido en [54]. A continuación, se muestra la información del protocolo RFTap capturada en uno de los paquetes, cuyo campo contiene una longitud de 32 bits.

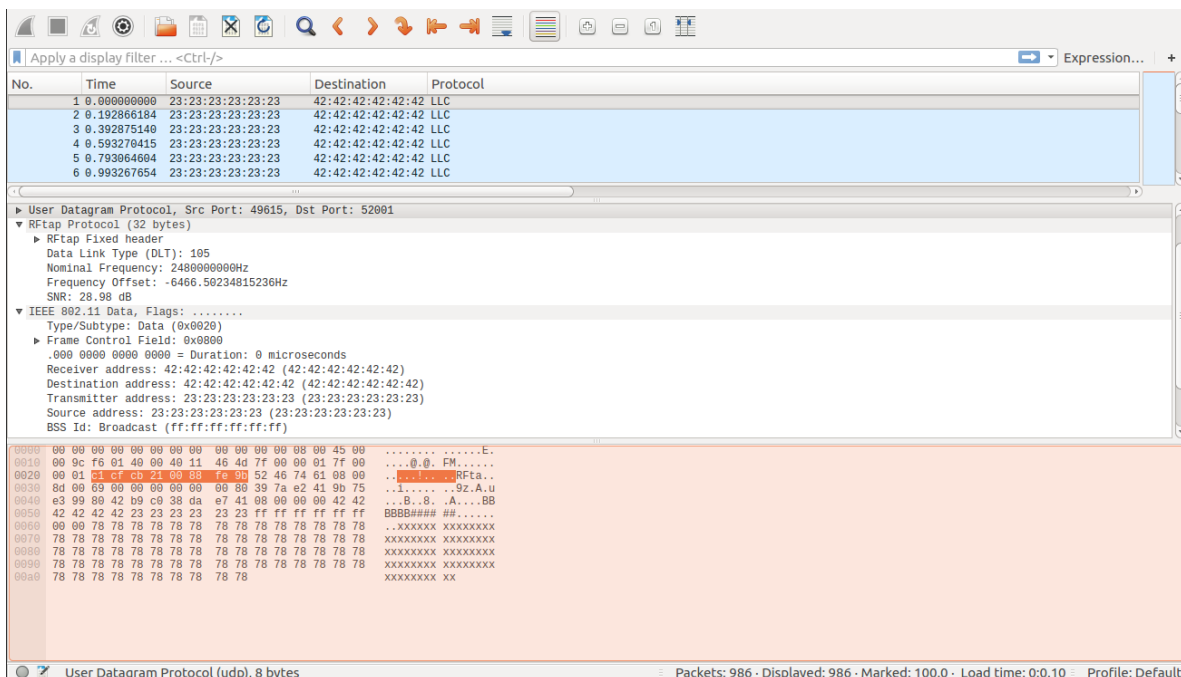


Fig 7.2 Captura de un paquete 802.11 con encapsulación RFTap.



Los datos luego son capturados en Wireshark y procesados utilizando la herramienta de software Matlab. Para determinar los resultados en cada uno de los casos se enviaron 1000 paquetes, de un total de 100 bytes (72 de payload), cada 200 ms. Cada una de las pruebas fueron realizadas cinco veces, y posteriormente se determinó el promedio de los resultados obtenidos en las mismas tales como: la razón de paquetes con error (PER, por sus siglas en inglés) determinado por la ecuación 7.1 en cada una de las corridas y el SNR recibido en cada uno de los paquetes. El BER se calculó utilizando posteriormente la ecuación 7.2 [56].

$$PER = 1 - \left(\frac{pkt_{rx}}{pkt_{tx}} \right) \quad (7.1)$$

$$1 - PER = (1 - BER)^L \quad (7.2)$$

donde:

- pkt_{rx} paquetes recibidos
- pkt_{tx} paquetes transmitidos
- L longitud del paquete en bytes

Básicamente, el PER es determinado por la longitud del paquete y la razón de error de bits (BER, por sus siglas en inglés). El BER es dependiente de la razón señal a ruido (SNR), el esquema de modulación y la codificación. Por su parte, el SNR se determina en relación a las pérdidas por trayectoria y las condiciones del canal. En redes inalámbricas móviles, las pérdidas por trayectoria, el desvanecimiento y la interferencia causan variaciones en el SNR recibido, el cual influencia la razón de error de bits. Por ende, la razón de error de paquetes tendrá una variabilidad correspondiente [56], la cual se define en la ecuación 7.2 presentada anteriormente.

El programa utilizado en MATLAB para la muestra de los resultados se encuentra en el anexo C.

7.2 Especificaciones del sistema

Las características y parámetros del sistema de transmisión se muestran en la Tabla 7.1, mientras que las características y parámetros del sistema de recepción se muestran en la tabla 7.2.

Tabla 7.1 Parámetros del sistema de transmisión.

Equipo	Componentes/Característica	Tipo
USRP	Modelo	N210
	Tarjeta hija	SBX-40MHz (400-4400 MHz)
	FPGA	Spartan 3A-DSP 3400
	Tamaño de muestras	14/16 bit
	Razón de muestreo	100/400 MS/s

	Interfaz	1GE
PC	Procesador	Core i3-4160 CPU @ 3.60GHz x 4
	Memoria RAM	8GB
	Disco Duro	500 GB
	Sistema Operativo	Ubuntu 16.04
	Software	GNU Radio
	Versión UHD	UHD_003_011

Tabla 7.2 Parámetros de sistema de recepción.

Equipo	Componentes/Característica	Tipo
USRP	Modelo	X310
	Tarjeta hija	CBX-120 MHz
	FPGA	Kintex 7-410T
	Tamaño de muestras	14/16 bit
	Razón de muestreo	200/800 MS/s
	Interfaz	1GE
PC	Procesador	Core i7-4770K CPU @ 3.50GHz x 8
	Memoria RAM	16GB
	Disco Duro	1 TB
	Sistema Operativo	Ubuntu 16.04
	Software	GNU Radio
	Versión UHD	UHD_003_011

En la implementación, se utilizó una antena de tipo omnidireccional modelo VERT2450 (figura 7.3) cuya frecuencia de operación es d 2.4 – 2.48 GHz y de 4.9 a 5.9 GHz, con una ganancia de 3 dBi.



Fig 7.3 Antena VERT2450.



Como se puede observar en las tablas 7.1 y 7.2, se utilizan dos tipos de tarjetas hijas, la CBX utilizada en recepción con el USRP X310 tiene un rango de operación de 1.2 a 6 GHz y la SBX utilizada en transmisión con los USRP N210 tiene un rango de operación de 400 a 4400 MHz. Por ende, debido a tales limitaciones de hardware en las tarjetas hijas compatibles con los USRP N210 no se puede utilizar la frecuencia de operación de 5.9 GHz establecida para el estándar 802.11p.

Tomando en cuenta la frecuencia de operación de las tarjetas hijas y el modelo de antena utilizado, se encuentra como factor común la banda de frecuencias de 2.4 GHz utilizadas por los sistemas WiFi; tratando de mitigar en mayor medida la gran cantidad de interferencias presentes en la misma, se decide utilizar una frecuencia soportada por las antenas y que se encuentre más cercana al canal 14 del espectro de frecuencias WiFi ya que constituye el extremo más alto del mismo y, prácticamente, no se utiliza. El canal 14 presenta una frecuencia central de 2.484 MHz, por ende, se decide ocupar la frecuencia 2.48GHz.

En la figura 7.4 se muestra la ocupación de los canales en la banda WiFi. Para ello se utiliza el Software Acrylic Wi-Fi Professional v3.2.6355.21915 [55] bajo licencia educacional asociado al correo con dominio @comunidad.unam.mx.

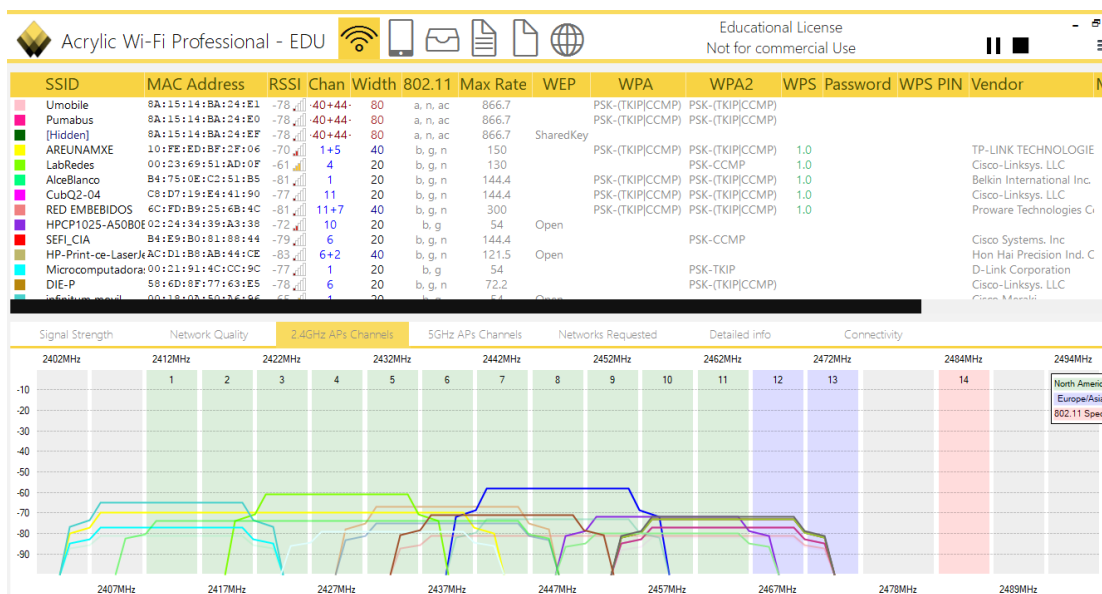


Fig 7.4 Ocupación de canales en la banda 2.4 GHz en el entorno de prueba.

El ancho de banda ocupado fue de 10 MHz según el establecido por el estándar. En los escenarios propuestos se utilizaron las modulaciones BPSK 1/2, QPSK 1/2, 16 QAM 1/2 y 64 QAM 2/3.

Debido a que de las 48 portadoras de datos, se tomaron 4 portadoras adicionales nulas para realizar una mejor estimación de canal, la tasa de transmisión del sistema MIMO fue modificada y está dada por la ecuación 7.3:



$$Data Rate = \frac{1}{T_s} * cc * m * N_{FFTdata} \quad (7.3)$$

Donde:

cc es la tasa de codificación

m es el número de bits por símbolo

$N_{FFTdata}$ es el número de portadoras de datos

T_s es el tiempo de símbolo OFDM y está dado a su vez por las ecuaciones 7.4 y 7.5.

$$\Delta f = \frac{BW}{N_{FFT}} = \frac{10MHz}{64} = 156250 \text{ Hz} \quad (7.4)$$

$$T_s = 1.25 \left(\frac{1}{\Delta f} \right) = 8 * 10^{-6} \quad (7.5)$$

Donde:

- Δf es la separación entre portadoras
- **BW** es el ancho de banda
- N_{FFT} es el número total de portadoras

Por ende, para los esquemas de modulación utilizados, se definen la tasa de transmisión como a continuación se muestra:

$$Data Rate_{BPSK \ 1/2} = \frac{1}{8*10^{-6}} * \frac{1}{2} * 1 * 44 = 2.75 \text{ Mbps} \quad (7.6)$$

$$Data Rate_{QPSK \ 1/2} = \frac{1}{8*10^{-6}} * \frac{1}{2} * 2 * 44 = 5.5 \text{ Mbps} \quad (7.7)$$

$$Data Rate_{16QAM \ 1/2} = \frac{1}{8*10^{-6}} * \frac{1}{2} * 4 * 44 = 11 \text{ Mbps} \quad (7.8)$$

$$Data Rate_{64QAM \ 2/3} = \frac{1}{8*10^{-6}} * \frac{2}{3} * 6 * 44 = 22 \text{ Mbps} \quad (7.9)$$

7.2.1 Escenario 1

La comunicación inalámbrica para este escenario se llevó a cabo en un lugar cerrado (*indoors*), en el Laboratorio de Electromagnetismo Aplicado (Q303) del Edificio Q “Luis G. Valdés Vallejo” de la Universidad Nacional Autónoma de México. La figura 7.5 muestra el escenario, y el objetivo trazado

fue medir el número de bits en error (BER), en función de la relación señal a ruido (SNR) medida en el receptor.

Para lograr diferentes valores de SNR, se realizó una variación de ganancia en el transmisor y el receptor en un rango de 10 dB estableciendo una distancia de 1m entre los dispositivos.



Fig 7.5 Primer escenario para la prueba del sistema MIMO.

Los resultados obtenidos se muestran a continuación tabulados respecto al promedio de los paquetes recibidos en cada prueba (tablas 7.3 – 7.7) y graficados de acuerdo al BER en relación con la distancia en metros (figuras 7.6 - 7.9).

a) BPSK ½ :

Tabla 7.3 Promedio de paquetes decodificados correctamente en relación al promedio del SNR recibido para BPSK ½.

SNR (dB)	Paquetes Rx SISO	SNR (dB)	Paquetes Rx MIMO
2.71896	1	2.21896	2
3.78801456	100	2.98801456	48.7296322
4.91468137	812	3.91468137	897.540852
7.67716606	995.004453	5.17716606	994.0098911
13.4801175	996	12.4801175	998.4482521
15.6523	999.4999998	14.38888	1000

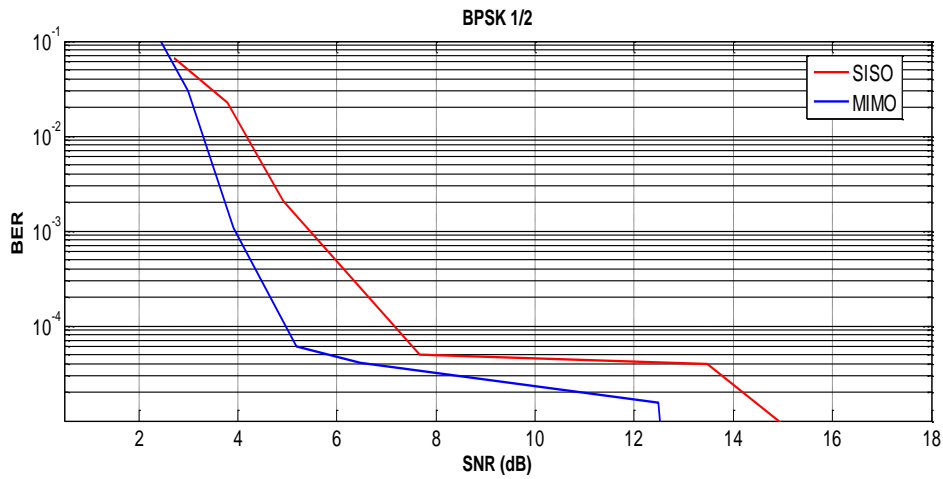


Fig 7.6 Gráfica de BER vs SNR para modulación BPSK 1/2.

b) QPSK 1/2:

Tabla 7.4 Promedio de paquetes decodificados correctamente en relación al promedio del SNR recibido para QPSK 1/2.

SNR	Paquetes Rx SISO	SNR	Paquetes Rx MIMO
6.11571515	66	5.75571515	23.4949425
6.78413297	192	6.48413297	260.349008
7.80022348	624.5	8.22234834	843.798503
10.497389	952	8.697389	942.522333
13.5404703	993	10.5404703	991.015826
16.6531315	996	15.6531315	997.994053
19.6825261	997	19.6825261	998.996035

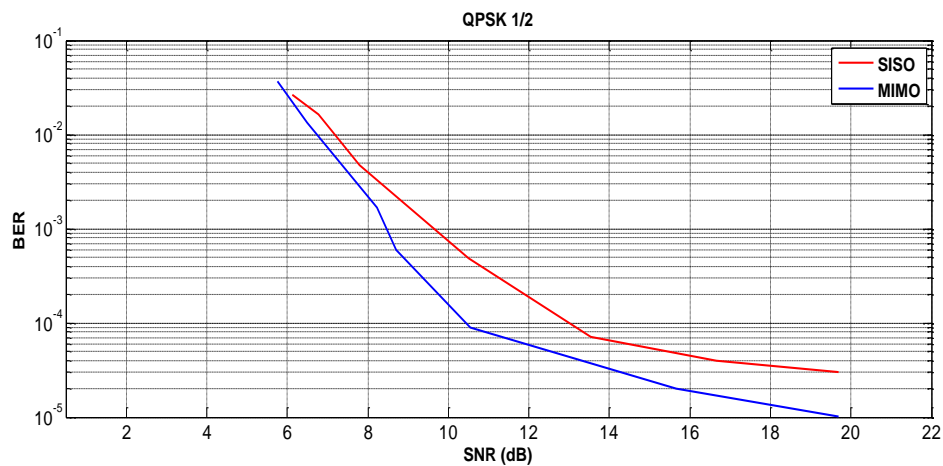


Fig 7.7 Gráfica de BER vs SNR para modulación QPSK 1/2.



c) 16 QAM 1/2:

Tabla 7.5 Promedio de paquetes decodificados correctamente en relación al promedio del SNR recibido para 16QAM 1/2.

SNR	Paquetes Rx SISO	SNR	Paquetes Rx MIMO
12.303643	99	11.803643	55.7216995
13.6646953	530	12.5046953	479.234211
16.0143	740	13.3292955	700.129327
16.7292955	774	16.2224709	805.995481
19.1362249	783	19.0224709	818.202117

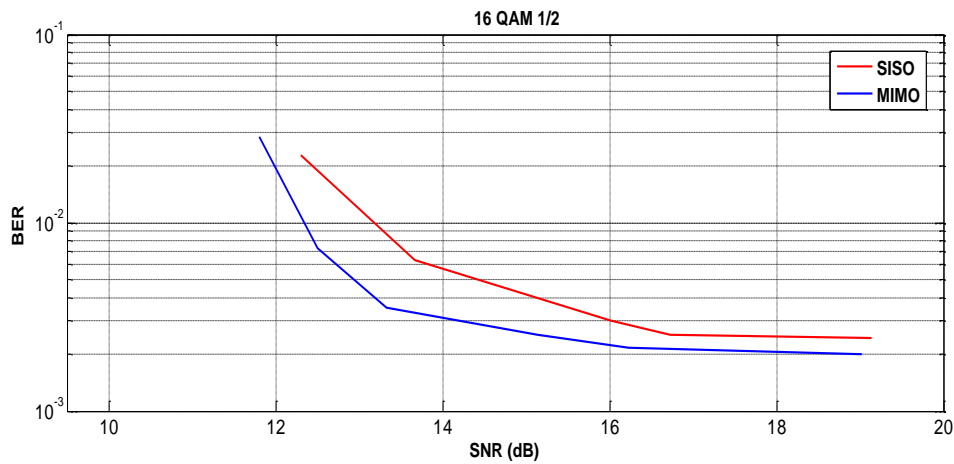


Fig 7.8 Gráfica de BER vs SNR para modulación 16QAM 1/2.

d) 64 QAM 2/3:

Tabla 7.6 Promedio de paquetes decodificados correctamente en relación al promedio del SNR recibido para 64 QAM 2/3.

SNR	Paquetes Rx SISO	SNR	Paquetes Rx MIMO
19.7312333	12	19.33133	3.88251945
22.4845504	278	20.9045504	100.441243
24.5169037	372	21.9169037	407.380224
26.3879995	468	25.3879995	517.588095
28.4648297	472.5	28.4548297	533.184629

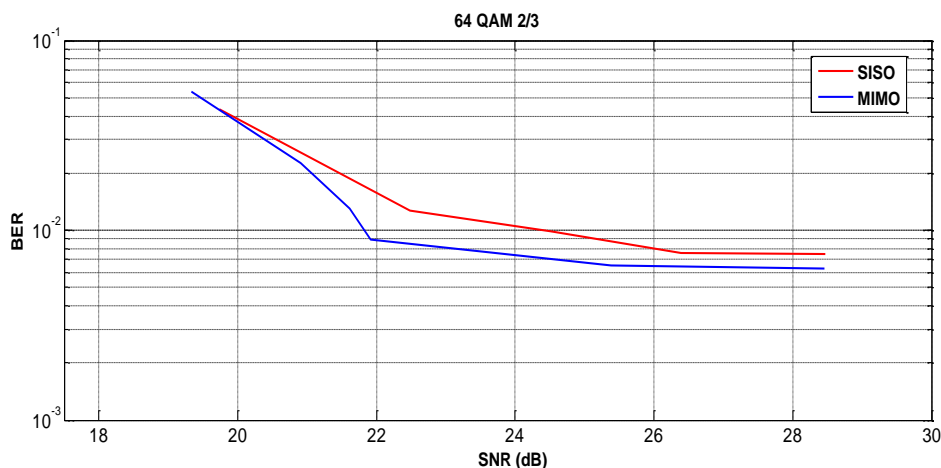


Fig 7.9 Gráfica de BER vs SNR para modulación 64 QAM 2/3.

En cada una de las gráficas para las distintas modulaciones utilizadas en este primer escenario, se pueden observar con el análisis de BER vs SNR en función de la cantidad de paquetes recibidos, que se comportan dentro del margen previsto según la teoría, dando resultados coherentes con una ganancia en diversidad de orden 2. Para todos los casos el sistema MIMO logró una mejora de hasta 3 dB, aunque para altos valores de SNR se mantienen los índices prácticamente constantes, debido a que por las condiciones del entorno y de procesamiento de la señal no se logra mejor recepción, excepto para BPSK1/2 que logra llegar a un 100% de recepción en 14 dB aproximadamente para el sistema MIMO.

7.2.2 Escenario 2

El segundo escenario establecido fue también en un ambiente de interiores, el objetivo fue determinar el comportamiento del BER vs la distancia entre el equipo transmisor y el equipo receptor. Los resultados se tomaron variando la posición del transmisor, y el escenario se ubicó entre el pasillo y el Laboratorio de Electromagnetismo Aplicado (Q303) del edificio Q “Luis G. Valdés Vallejo” de la Universidad Nacional Autónoma de México. El escenario logró abarcar un aproximado de 24 metros de largo y los equipos USRP fueron ubicados a una altura de 1m. En la figura 7.10 izquierda se observa el sistema SISO tradicional y a la derecha se observa el sistema MIMO propuesto.



Fig 7.10 Implementación en hardware del sistema SISO (izquierda) y sistema MISO (derecha).

Los resultados obtenidos se muestran a continuación tabulados respecto al promedio de los paquetes recibidos en cada prueba (tablas 7.7 – 7.10) y graficados de acuerdo al BER en relación con la distancia en metros (figuras 7.11 – 7.14).

a) BPSK $\frac{1}{2}$:

Tabla 7.7 Promedio de paquetes recibidos en relación a la distancia para BPSK 1/2.

Distancia (m)	Paquetes Rx SISO	Paquetes Rx MIMO
2	1000	1000
4	1000	1000
6	1000	1000
8	1000	1000
10	1000	1000
12	999	1000
14	1000	1000
16	1000	1000
18	1000	1000
20	995.5	1000
22	993.3333333	995.5
24	999.5	999.3333

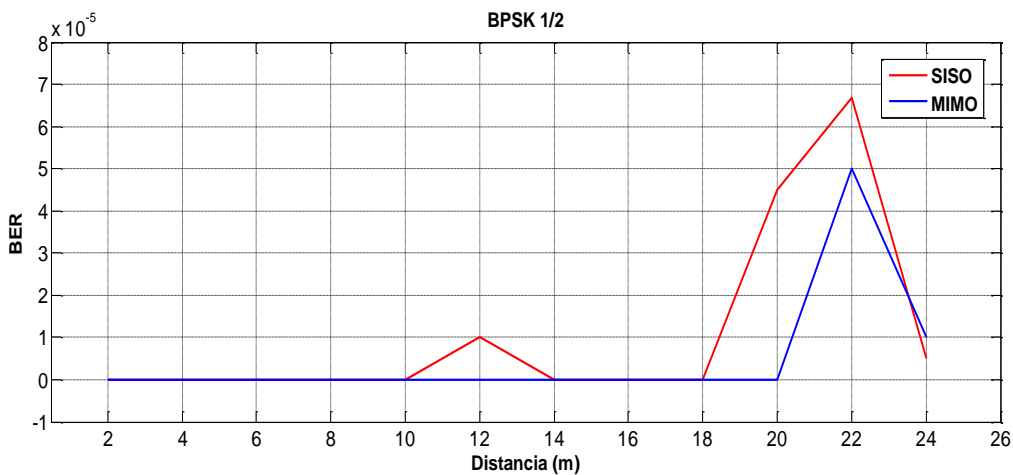


Fig 7.11 Gráfica de BER vs distancia para modulación BPSK 1/2.

b) QPSK1/2:

Tabla 7.8 Promedio de paquetes recibidos en relación a la distancia para QPSK 1/2.

Distancia (m)	Paquetes Rx SISO	Paquetes Rx MIMO
2	994.75	1000
4	997	1000
6	998	1000
8	999	1000
10	995	1000
12	994	999
14	998	999
16	992	997
18	991.666667	996.9999
20	984.666667	988.888
22	980	981
24	993.666667	995

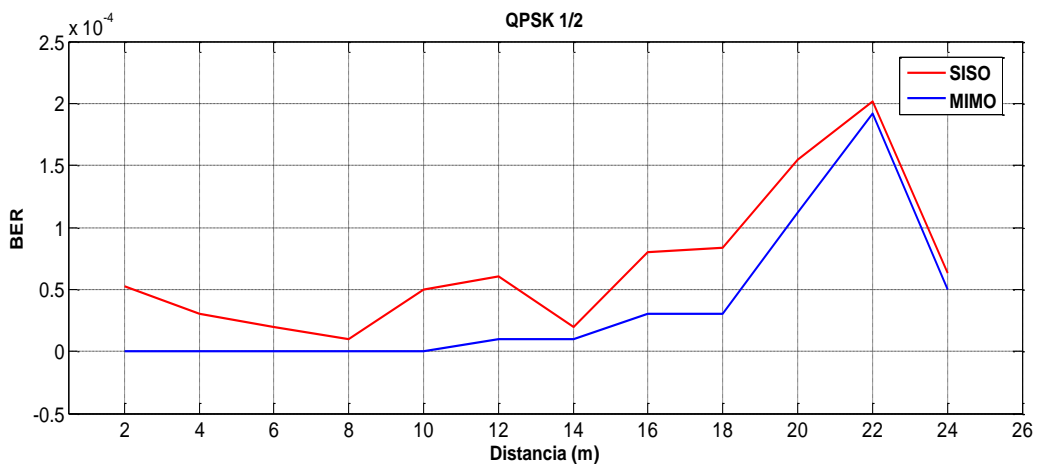


Fig 7.12 Gráfica de BER vs distancia para modulación QPSK 1/2.

c) 16QAM 1/2:

Tabla 7.9 Promedio de paquetes recibidos en relación a la distancia para 16 QAM 1/2.

Distancia (m)	Paquetes Rx SISO	Paquetes Rx MIMO
2	781.428571	880
4	776.333333	840.75
6	730	803.33333
8	773.75	798
10	771.333333	797
12	767.333333	779.33333
14	761.333333	780
16	759	763
18	728	757.5
20	754	758
22	761.333333	770
24	650	759

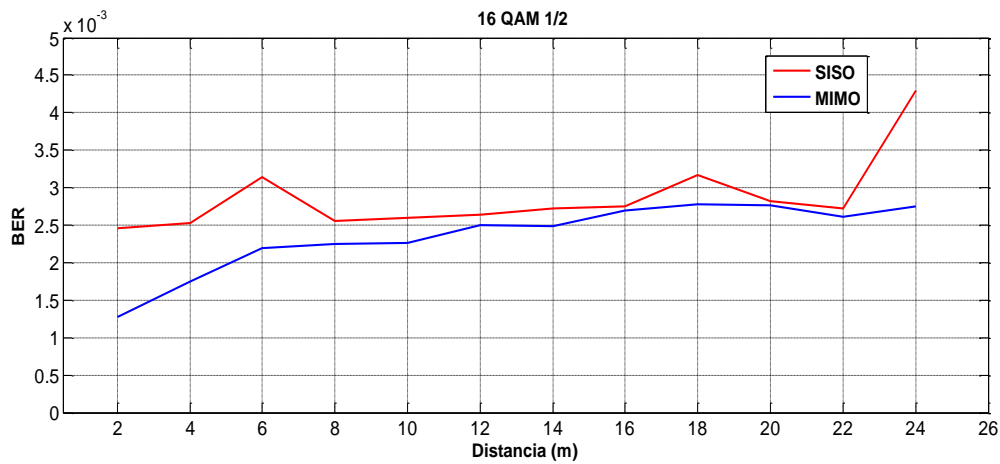


Fig 7.13 Gráfica de BER vs distancia para modulación 16 QAM 1/2.

d) 64QAM 2/3:

Tabla 7.10 Promedio de paquetes recibidos en relación a la distancia para 64 QAM 2/3.

Distancia (m)	Paquetes Rx SISO	Paquetes Rx MIMO
2	133.666667	197
4	49.8	99.6666667
6	79	102
8	28	109
10	112	189.8
12	91.6666667	140
14	3.66666667	19.4
16	4	5
18	0	2

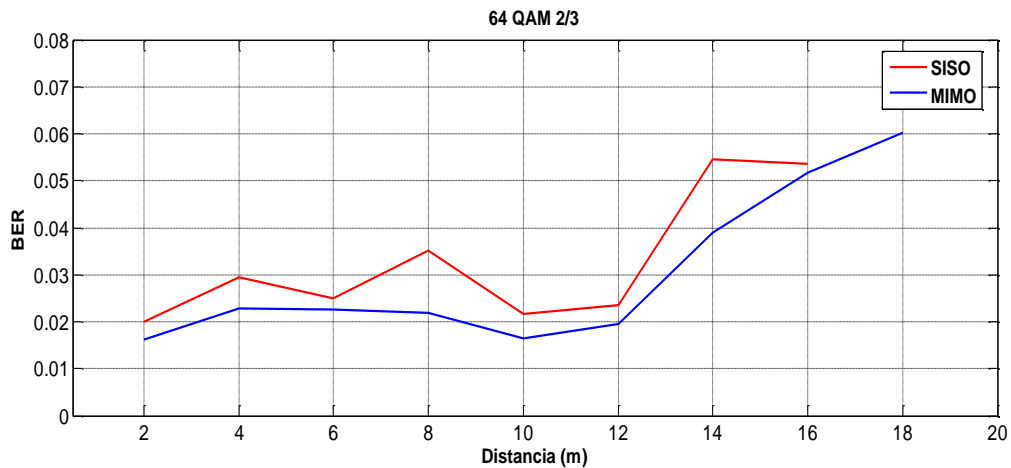


Fig 7.14 Gráfica de BER vs distancia para modulación 64 QAM 2/3.

Como se observa en las gráficas anteriores, los resultados prácticos de BER vs Distancia no se encuentran en total congruencia con la teoría, ya que en algunos casos se muestra mejor BER a mayor distancia. Debido a que los experimentos fueron realizados en ambientes estrechos y llenos de aparatos electrónicos, se supone que la influencia de éstos sobre la transmisión y el efecto de las multitrayectorias pudieron afectar los resultados obtenidos. Éste es uno de los principales problemas que conlleva el manejo de la familia USRP; son sumamente susceptibles a los efectos generados por multitrayectorias. Durante los experimentos se definieron las mayores variaciones en los resultados alrededor de los 6, 12, 18 y 24 metros principalmente, tales resultados corresponden con: el medio del laboratorio Q303, la entrada del pasillo al laboratorio, el medio del pasillo y el final del pasillo, respectivamente.

Tomando en cuenta los resultados de distancia tomados en este segundo escenario, se logró alcanzar una recepción en la máxima distancia comprendida de 24 metros para todos los casos excepto en la modulación 64 QAM 2/3 donde se logró recepción hasta un máximo de 18 metros con el sistema MIMO, logrando una mejora de 2 metros con respecto al sistema SISO que logró recepcionar hasta un aproximado de 16 metros.



CONCLUSIONES

En el presente trabajo de tesis se implementaron de manera práctica la capa física de un sistema tradicional bajo el estándar 802.11p. Se describió la funcionalidad de cada bloque que rige el sistema tanto en transmisión como en recepción, y se analizó el funcionamiento de los mismos. Posteriormente se llevó a cabo la implementación del mismo sistema, pero adicionando una estructura MIMO con codificación de Alamouti utilizando dos antenas en transmisión y una en recepción logrando una ganancia por diversidad en el orden de 2 para el mismo.

El presente trabajo de tesis destacó el increíble potencial de los radios definidos por software, comprobando una vez más que las plataformas SDR permiten la implementación de la capa física de prácticamente cualquier sistema de comunicaciones sin realizar cambios en hardware e incluyendo funciones necesarias como modulación, codificación, ancho de banda y demás, incluidas en los estándares.

Según el estado del arte para la presente investigación, se decidió realizar ciertas modificaciones a la capa física del estándar 802.11p a un entorno multiantena con dos antenas en transmisión y una en recepción, para comprobar las grandes ventajas que podría proveer éste tipo de sistema en un ambiente vehicular, donde la robustez de la transmisión constituya un tópico esencial. El sistema constituye, hasta donde se tiene conocimiento, la primera implementación del estándar utilizando la técnica MIMO y la codificación Alamouti en GNU Radio y soportando los esquemas de modulación especificados (hasta 64 QAM).

De los resultados obtenidos, es importante destacar una reducción del 8.33% en la tasa de transmisión de bits para todas las modulaciones, debido a que, para realizar una correcta estimación de canal para los dos canales recibidos, fue necesario la incorporación de cuatro portadoras nulas para transmitir por cada antena, mientras que por la otra se transmitía una portadora piloto y así eliminar la interferencia entre las mismas. Lo mismo conlleva a tomar en cuenta la relación de compromiso existente entre un seguimiento más exacto de la estimación de canal al insertar símbolos pilotos intercalados entre los símbolos de datos y la razón de transmisión de bits.

El sistema fue implementado en dos escenarios distintos en interiores realizándose un análisis comparativo entre el sistema tradicional de una sola entrada y una sola salida (SISO) con el sistema MIMO 2x1 propuesto. Tales pruebas confirmaron la robustez de BPSK, siendo en todos los casos la de menos BER y la elevada velocidad transmisión que se puede alcanzar con 64 QAM, aunque con alto BER.

Del primer escenario en interiores se puede concluir que el comportamiento de BER vs SNR en los esquemas de modulación y codificación se comportaron dentro del margen previsto según la teoría, dando resultados coherentes con una ganancia en diversidad de orden 2. Para todos los casos el sistema MIMO logró una mejora de hasta 3 dB, aunque para altos valores de SNR se mantienen los



índices prácticamente constantes, debido a que por las condiciones del entorno y de procesamiento de la señal no se logra mejor recepción, excepto para BPSK1/2 que logra llegar a un 100% de recepción en 14 dB aproximadamente para el sistema MIMO.

Del segundo escenario cabe destacar que algunos resultados no eran congruentes con la teoría debido al multitrayecto presente en el entorno de pruebas. El principal problema se observó en los resultados obtenidos del sistema SISO, ya que en algunos casos se lograba mejor BER a mayor distancia, presentando una elevada variabilidad en la toma de muestras. Dicho fenómeno se observó en menor medida en el sistema MIMO, donde, además de lograr una mejora en el BER tomando en cuenta una misma distancia, confirmó que puede dotar al sistema de comunicaciones de mayor robustez frente al fenómeno del multitrayecto, al menos en ambientes interiores. Por ende, para una realización más exacta de cualquier tipo de experimentos que hagan uso de los radios USRP, el ambiente de pruebas debe ser implementado en un entorno menos susceptible al efecto de las multitrayectorias. Para la presente investigación, los dispositivos USRP necesitaban ser conectados a través de una interfaz Gigabit Ethernet por lo que se utilizaron computadoras de escritorio que no eran para nada portables. Debido a lo anterior, el pasillo resultó ser la opción más viable para la toma de datos.

Tomando en cuenta los resultados de distancia tomados en el segundo escenario, se logró alcanzar una recepción en la máxima distancia comprendida de 24m para todos los casos excepto en la modulación 64 QAM 2/3 donde se logró recepción hasta un máximo de 18 metros con el sistema MIMO, logrando una mejora de 2m con respecto al sistema SISO que logró recibir hasta un aproximado de 16 metros.

A manera general se puede concluir que la plataforma SDR permite implementar y evaluar diversas tecnologías de capa física sin necesidad de cambiar el hardware y solamente con la modificación del software. Aunque los dispositivos USRP presentan algunas limitaciones para el procesamiento en tiempo real, éstos han permitido la implementación de prototipos coherentes con los sistemas reales que han servido de base tanto en entornos académicos como investigativos.

Dichas limitaciones, serán cada vez más aminoradas con el desarrollo de las plataformas SDR, las PCs y las interfaces de interconexión entre los mismos.

Por otra parte, aunque la utilización del estándar IEEE 802.11p presenta algunas limitantes que están siendo superadas por la tecnología LTE (*Long Term Evolution*) en sistemas ITS, la implementación de la tecnología MIMO realizada en la presente investigación sirve como base para el estudio de cualquier sistema que haga uso de la misma, entre los que se pueden mencionar los estándares WLAN y el mismo LTE.



TRABAJO FUTURO

El desarrollo del presente trabajo proporciona un amplio panorama en cuanto al diseño e implementación de los sistemas MIMO utilizando plataformas SDR. Dicha tecnología se encuentra aún en desarrollo permitiendo ser investigada con prototipos cuyas funciones y configuraciones pueden ser modificadas a través de software y por tanto, se propone:

- Realizar la implementación de otras técnicas de codificación STBC y compararlas con la codificación de Alamouti ya implementada.
- Extender el sistema a dos antenas transmisoras y dos antenas receptoras (2x2) utilizando la misma codificación de Alamouti, y de la misma manera compararlo con otras técnicas de codificación.
- Implementar diversas técnicas de estimación de canal y deducir la más adecuada para un sistema MIMO utilizado en ambientes vehiculares, tomando en cuenta fenómenos como desvanecimientos rápidos y efecto Doppler.



REFERENCIAS

- [1] J. Mitola, "Software radios-survey, critical evaluation and future directions," in *Telesystems Conference, 1992. NTC-92., National*, 1992, pp. 13/15-13/23.
- [2] F. Dressler, F. Kargl, J. Ott, O. K. Tonguz, and L. Wischhof, "Research challenges in intervehicular communication: lessons of the 2010 Dagstuhl Seminar," *IEEE Communications Magazine*, vol. 49, 2011.
- [3] S. R. C. Shah, S. Roy, and W. Brunette, "Standard specification for telecommunications and information exchange between roadside and vehicle systems-5 GHz band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 2003.
- [4] C. M. A. Morales, "Implementación de un Receptor OFDM IEEE 802.11 basado en SDR," Master, Universidad Nacional Autónoma de México, México, 2015.
- [5] G. R. L. Portilla, "Implementación de un Transmisor 802.11 OFDM basado en SDR," Master, Universidad Nacional Autónoma de México, México, 2015.
- [6] P. Fuxjäger, A. Costantini, D. Valerio, P. Castiglione, G. Zacheo, T. Zemen, and F. Ricciato, "IEEE 802.11 p transmission using GNURadio," in *6th Karlsruhe Workshop on Software Radios (WSR)*, 2010, pp. 1-4.
- [7] "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Redline," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999) - Redline*, pp. 1-1238, 2007.
- [8] J. R. Gutierrez-Agullo, B. Coll-Perales, and J. Gozalvez, "An IEEE 802.11 MAC Software Defined Radio implementation for experimental wireless communications and networking research," in *Wireless Days (WD), 2010 IFIP*, 2010, pp. 1-5.
- [9] C. Schmidt-Knorreck, D. Knorreck, and R. Knopp, "IEEE 802.11 p Receiver Design for Software Defined Radio Platforms," in *Digital System Design (DSD), 2012 15th Euromicro Conference on*, 2012, pp. 437-444.
- [10] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, "An IEEE 802.11 a/g/p OFDM receiver for GNU radio," in *Proceedings of the second workshop on Software radio implementation forum*, 2013, pp. 9-16.
- [11] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, "Towards an open source IEEE 802.11 p stack: a full sdr-based transceiver in gnu radio," in *Vehicular Networking Conference (VNC), 2013 IEEE*, 2013, pp. 143-149.
- [12] S. Poochaya, P. Uthansakul, M. Uthansakul, and O. Tonguz, "THE IMPROVEMENT OF A VEHICLE TO INFRASTRUCTURE COMMUNICATION LINK FOR ITS/DSRC USING SDR AND MIMO TECHNOLOGY," *Suranaree Journal of Science & Technology*, vol. 21, 2014.
- [13] S. Han, K. Zhao, L.-Q. Yang, and X. Cheng, "Performance evaluation for multi-antenna vehicular communication based on IEEE 802.11 p standard," in *Computing, Networking and Communications (ICNC), 2016 International Conference on*, 2016, pp. 1-5.
- [14] S. Eichler, "Performance evaluation of the IEEE 802.11 p WAVE communication standard," in *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, 2007, pp. 2199-2203.
- [15] T. Group, "Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems-5GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications," ed: ASTM E2213-03, 2003.



- [16] I. S. Association, "802.11-2012-IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std*, vol. 802, 2012.
- [17] D. Jiang and L. Delgrossi, "IEEE 802.11 p: Towards an international standard for wireless access in vehicular environments," in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, 2008, pp. 2036-2040.
- [18] A. Sassi, F. Charfi, L. Kamoun, Y. Elhillali, and A. Rivenq, "OFDM transmission performance evaluation in V2X Communication," *arXiv preprint arXiv:1410.8039*, 2014.
- [19] A. Abdelgader and W. Lenan, "The physical layer of the IEEE 802.11 p WAVE communication standard: the specifications and challenges," in *Proceedings of the World Congress on Engineering and Computer Science*, 2014, p. 71.
- [20] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and H. V. Poor, *MIMO wireless communications*: Cambridge university press, 2007.
- [21] S. Moser, L. Behrendt, and F. Slomka, "MIMO-enabling PHY layer enhancement for vehicular ad-hoc networks," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2015 IEEE*, 2015, pp. 142-147.
- [22] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on selected areas in communications*, vol. 16, pp. 1451-1458, 1998.
- [23] R. Negi and J. Cioffi, "Pilot tone selection for channel estimation in a mobile OFDM system," *IEEE Transactions on Consumer Electronics*, vol. 44, pp. 1122-1128, 1998.
- [24] M. K. Ozdemir and H. Arslan, "Channel estimation for wireless OFDM systems," *IEEE Communications surveys & tutorials*, vol. 9, pp. 18-48, 2007.
- [25] J. K. Cavers, "Pilot symbol assisted modulation and differential detection in fading and delay spread," *IEEE transactions on communications*, vol. 43, pp. 2206-2212, 1995.
- [26] Y. Li, "Pilot-symbol-aided channel estimation for OFDM in wireless systems," *IEEE transactions on vehicular technology*, vol. 49, pp. 1207-1215, 2000.
- [27] S. Coleri, M. Ergen, A. Puri, and A. Bahai, "Channel estimation techniques based on pilot arrangement in OFDM systems," *IEEE Transactions on broadcasting*, vol. 48, pp. 223-229, 2002.
- [28] M. Engels, *Wireless OFDM systems: How to make them work?:* Springer Science & Business Media, 2002.
- [29] J. A. Fernandez, D. D. Stancil, and F. Bai, "Dynamic channel equalization for IEEE 802.11 p waveforms in the vehicle-to-vehicle channel," in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, 2010, pp. 542-551.
- [30] B. Bloessl, M. Gerla, and F. Dressler, "IEEE802. 11p in fast fading scenarios: from traces to comparative studies of receive algorithms," in *Proceedings of the First ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services*, 2016, pp. 1-5.
- [31] A. Marwanto, M. A. Sarijari, N. Fisal, S. K. S. Yusof, and R. A. Rashid, "Experimental study of OFDM implementation utilizing GNU Radio and USRP-SDR," in *Communications (MICC), 2009 IEEE 9th Malaysia International Conference on*, 2009, pp. 132-135.
- [32] Mathworks. (12/05/2017). Available: <https://es.mathworks.com/discovery/sdr.html>
- [33] H.-H. Cho, C.-F. Lai, T. K. Shih, and H.-C. Chao, "Integration of SDR and SDN for 5G," *IEEE Access*, vol. 2, pp. 1196-1204, 2014.
- [34] M. Srilatha, R. Hemalatha, T. S. Aditya, and P. A. Sravanthi, "Knowledge based analysis of software defined radio for wireless communication: A preliminary survey," *International Journal of Computer Applications*, vol. 71, 2013.



- [35] M. Föhnle, "Software-Defined Radio with GNU Radio and USRP/2 hardware frontend: setup and FM/GSM applications," *Hochschule Ulm University of Applied Sciences, Institute of Communication Technology*, 2010.
- [36] E. Research. (10/11/2016). *USRP N210*. Available: <https://www.ettus.com/product/details/UN210-KIT>
- [37] E. Research. *USRP™ n200/n210 networked series*. Available: https://www.ettus.com/content/files/07495_Ettus_N200-210_DS_Flyer_HR_1.pdf
- [38] E. Research. *USRP X310*. Available: <https://www.ettus.com/product/details/X310-KIT>
- [39] E. Research, "SBX 400-4400 MHz Rx/Tx (40 MHz)."
- [40] E. Research. *CBX 1200-6000 MHz Rx/Tx (120 MHz, X Series only)*. Available: <https://www.ettus.com/product/details/CBX120>
- [41] A. Verduin, "GNU Radio wireless protocol analysis approach," *Master Program System and Network Engineering, Universiteit Van Amsterdam*, 2008.
- [42] *Guided Tutorial GRC*. Available: https://wiki.gnuradio.org/index.php/Guided_Tutorial_GRC
- [43] G. Radio, "QT Graphical User Interface."
- [44] G. Radio. *Digital Modulation*. Available: https://gnuradio.org/doc/doxygen/page_digital.html
- [45] G. Radio. *OFDM*. Available: https://gnuradio.org/doc/doxygen/page_ofdm.html
- [46] G. Radio. *Packet Data Transmission*. Available: https://gnuradio.org/doc/doxygen/page_packet_data.html
- [47] E. Research, "Application Note. Frontends, Sub-Device Specifications, and Antenna Port Selection.."
- [48] T. Rondeau, N. McCarthy, and T. O'Shea, "SIMD programming in GNURadio: Maintainable and user-friendly algorithm optimization with VOLK," in *Conference on Communications Technologies and Software Defined Radio (SDR'12). Brussels, Belgium: Wireless Innovation Forum Europe*, 2012.
- [49] C.-H. Liu, "On the design of OFDM signal detection algorithms for hardware implementation," in *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*, 2003, pp. 596-599.
- [50] E. Sourour, H. El-Ghoroury, and D. McNeill, "Frequency Offset Estimation and Correction in the IEEE 802.11 a WLAN," in *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, 2004, pp. 4923-4927.
- [51] G. Radio. (05/02/2017). Available: https://wiki.gnuradio.org/index.php/Guided_Tutorial_GNU_Radio_in_C%2B%2B
- [52] H. Zhongqiu and Y. Xing, "Pilot Aided Channel Estimation of MIMO OFDM Systems," in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*.
- [53] Github. *What is RFTap?* Available: <https://rftap.github.io/>
- [54] Tcpcap/Libpcap. (2000-2017). *TCPDUMP & Libcap*. Available: <http://www.tcpdump.org/linktypes.html>
- [55] T. S. SL, "Analizador WiFi avanzado: Acrylic WiFi Profesional," 2017.
- [56] YIN, Jun; WANG, Xiaodong; AGRAWAL, Dharma P. Optimal packet size in error-prone channel for IEEE 802.11 distributed coordination function. En *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*. IEEE, 2004. p. 1654-1659.

ANEXOS

Anexo A. Hoja de datos USRP N210



FEATURES:

- Use with GNU Radio, LabVIEW™ and Simulink™
- Modular Architecture: DC-6 GHz
- Dual 100 MS/s, 14-bit ADC
- Dual 400 MS/s, 16-bit DAC
- DDC/DUC with 25 MHz Resolution
- Up to 50 MS/s Gigabit Ethernet Streaming
- Fully-Coherent MIMO Capability
- Gigabit Ethernet Interface to Host
- 2 Gbps Expansion Interface
- Spartan 3A-DSP 1800 FPGA (N200)
- Spartan 3A-DSP 3400 FPGA (N210)
- 1 MB High-Speed SRAM
- Auxiliary Analog and Digital I/O
- 2.5 ppm TCXO Frequency Reference
- 0.01 ppm w/ GPSDO Option

N200/N210 PRODUCT OVERVIEW:

The Ettus Research™ USRP™ N200 and N210 are the highest performing class of hardware of the USRP™ (Universal Software Radio Peripheral) family of products, which enables engineers to rapidly design and implement powerful, flexible software radio systems. The N200 and N210 hardware is ideally suited for applications requiring high RF performance and great bandwidth. Such applications include physical layer prototyping, dynamic spectrum access and cognitive radio, spectrum monitoring, record and playback, and even networked sensor deployment.

The Networked Series products offers MIMO capability with high bandwidth and dynamic range. The Gigabit Ethernet interface serves as the connection between the N200/N210 and the host computer. This enables the user to realize 50 MS/s of real-time bandwidth in the receive and transmit directions, simultaneously (full duplex).

The Networked Series MIMO connection is located on the front panel of each unit. Two Networked Series units may be connected to realize a complete 2x2 MIMO configuration using the optional MIMO cable. External PPS and reference inputs can also be used to create larger multi-channel systems. The N200 and N210 are largely the same, except that the N210 features a larger FPGA for customers that intend to integrate custom FPGA functionality.

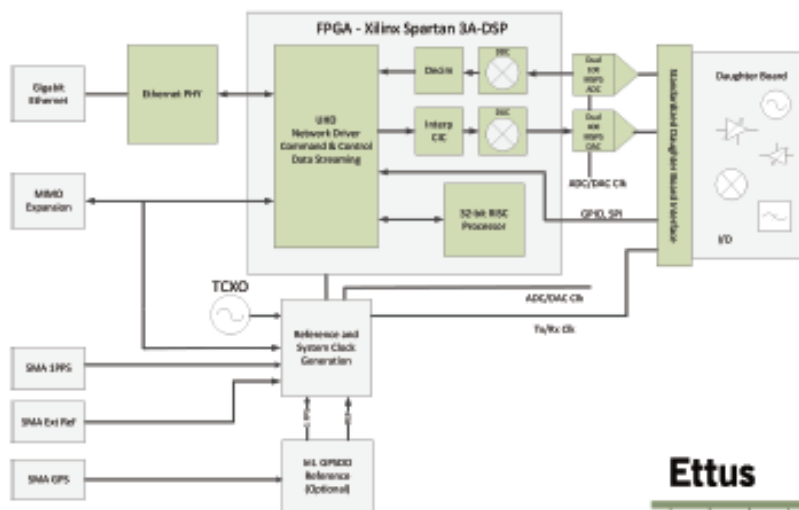
The USRP Hardware Driver™ is the official driver for all Ettus Research products. The USRP Hardware Driver supports Linux, Mac OSX, Windows.



SPECIFICATIONS

Spec	Typ.	Unit	Spec	Typ.	Unit
POWER			RF PERFORMANCE (W/ WBX)		
DC Input	6	V	SSB/LO Suppression	35/50	dBc
Current Consumption	1.3	A	Phase Noise (1.8 GHz)		
w/ WBX Daughterboard	2.3	A	10 kHz	-80	dBc/Hz
CONVERSION PERFORMANCE AND CLOCKS			100 kHz	-100	dBc/Hz
ADC Sample Rate	100	MS/s	1 MHz	-137	dBc/Hz
ADC Resolution	14	bits	Power Output	15	dBm
ADC Wideband SFDR	88	dBc	IIP3	0	dBm
DAC Sample Rate	400	MS/s	Receive Noise Figure	5	dB
DAC Resolution	16	bits	PHYSICAL		
DAC Wideband SFDR	80	dBc	Operating Temperature	0 to 55°	C
Host Sample Rate (8b/16b)	50/25	MS/s	Dimensions (l x w x h)	22 x 16 x 5	cm
Frequency Accuracy	2.5	ppm	Weight	1.2	kg
w/ GPSDO Reference	0.01	ppm			

* All specifications are subject to change without notice.



ABOUT ETTUS RESEARCH:

Ettus Research is an innovative provider of software defined radio hardware, including the original Universal Software Radio Peripheral (USRP) family of products. Ettus Research products maintain support from a variety of software frameworks, including GNU Radio. Ettus Research is a leader in the GNU Radio open-source community, and enables users worldwide to address a wide range of research, industry and defense applications. The company was founded in 2004 and is based in Mountain View, California. As of 2010, Ettus Research is a wholly owned subsidiary of National Instruments.

1043 North Shoreline Blvd
Suite 100
Mountain View, CA 94043
P 650.967.2870 www.ettus.com
F 866.807.9801

Anexo B. Hoja de datos USRP X310

FEATURES

- Two wideband RF daughterboard slots
 - Up 120MHz bandwidth per channel
 - Selection covers DC to 6 GHz
- Large, customizable Kintex-7 FPGA
 - USRP X300 - XC7K325T
 - USRP X310 - XC7K410T
- UHD architecture provides compatibility:
 - GNURadio
 - C++ API/Python
 - Other third-party frameworks & applications
- Multiple high-speed interfaces
 - Dual SFP(+) ports for 1/10 Gigabit Ethernet
 - PCIe x4
- Flexible clocking architecture
 - Configurable sample clock
 - Optional GPS-disciplined OCXO
 - Coherent operation with 10 MHz/1 PPS
- Compact and rugged half-wide 1U form factor



SAMPLE APPLICATIONS

- Advanced Wireless Prototyping (WiFi/Cellular)
- Massive MIMO Testbeds
- Passive RADAR
- Signals Intelligence

USRP X300 and X310 Product Overview

The Ettus Research USRP X300 and X310 are high-performance, scalable software defined radio (SDR) platforms for designing and deploying next generation wireless communications systems. The hardware architecture combines two extended bandwidth daughterboard slots covering DC – 6 GHz with up to 120 MHz of baseband bandwidth, multiple high-speed interface options (PCIe, Dual 1/10 Gige), and a large user-programmable Kintex-7 FPGA in a convenient desktop or rack-mountable half-wide 1U form factor. In addition to providing best-in-class hardware performance, the open source software architecture of the USRP X300 and X310 provides cross-platform UHD driver support making it compatible with a large number of USRP supported development frameworks, reference architectures, and open source projects.

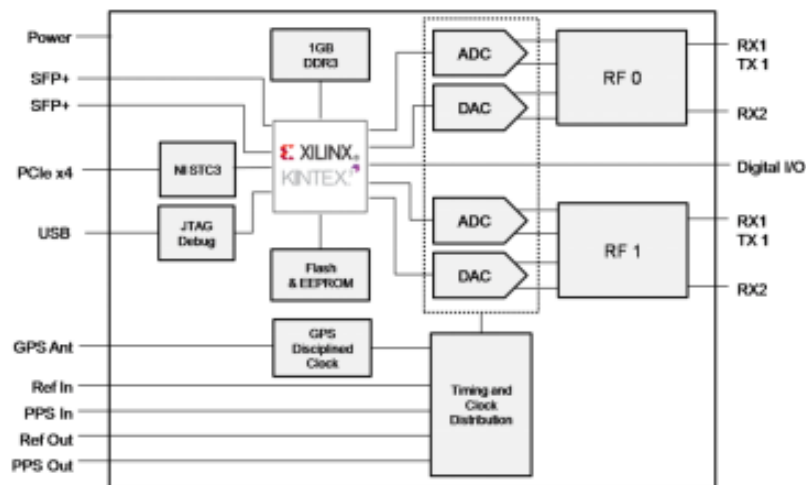


Spec	Typ.	Unit
Power		
DC Input	12	V
Power Consumption (2x SBX-120)	45	W
Conversion Performance and Clocks		
ADC Sample Rate (max)	200	MS/s
ADC Resolution	14	bits
DAC Sample Rate (max)	800	MS/s
DAC Resolution	16	bits
Host Sample Rate (16b) **	200	MS/s
Internal Reference Accuracy	2.5	ppm
Accuracy w/ GPSDO Option (not locked to GPS)	20	ppb

Spec	Typ.	Unit
RF Performance (with SBX-120)		
SSB/LO Suppression	-35/50	dBc
Phase Noise		
3.5 GHz	1.0	deg RMS
6 GHz	1.5	deg RMS
Power Output	>10	dBm
IIP3 (@ typ NF)	0	dBm
Typical Noise Figure	8	dB
Physical		
Dimensions (half-wide, 1U)	27.7 x 21.8 x 3.9	cm
Weight (w/ 2x SBX-120)	1.7	kg

*All specifications are subject to change without notice.

** Host sample rate dependent on selected interface and host-PC performance.



About Ettus Research

Ettus Research is an innovative provider of software defined radio hardware, including the original Universal Software Radio Peripheral (USRP) family of products. Ettus Research products maintain support from a variety of software frameworks, including GNU Radio. Ettus Research is a leader in the GNU Radio open-source community, and enables users worldwide to address a wide range of research, industry and defense applications. The company was founded in 2004 and is based in Mountain View, California. As of 2010, Ettus Research is a wholly owned subsidiary of National Instruments.

4600 Patrick Henry Dr.
Santa Clara, CA 95054
United States of America

P 650.967.2870 www.ettus.com
F 866.807.9801





Anexo C. Procesamiento de datos en Matlab tomados desde un libro Excel

%Yilena Abrahantes Hurtado
%Programacion para Procesamiento de Datos

clear all
clc

%%
%%

% Cargar los Datos Desde un Excel

[Tablas DIR]=uigetfile('* .xls','Cargar los Resultados de las Simulaciones:');

% Datos de los libros del Excel

pkt_rx1=xlsread(Tablas,'SISO');
pkt_rx2=xlsread(Tablas,'MIMO');

%%
%%

%Cargando los datos de SISO

Sim1=pkt_rx1(:,1);SNR1=pkt_rx1(:,2);Sim2=pkt_rx1(:,3);SNR2=pkt_rx1(:,4);Sim3=pkt_rx1(:,5);
SNR3=pkt_rx1(:,6);Sim4=pkt_rx1(:,7);SNR4=pkt_rx1(:,8);Sim5=pkt_rx1(:,9);SNR5=pkt_rx1(:,10);

%Cargando los datos de MIMO

Sim6=pkt_rx2(:,1);SNR6=pkt_rx2(:,2);Sim7=pkt_rx2(:,3);SNR7=pkt_rx2(:,4);Sim8=pkt_rx2(:,5);
SNR8=pkt_rx2(:,6);Sim9=pkt_rx2(:,7);SNR9=pkt_rx2(:,8);Sim10=pkt_rx2(:,9);SNR10=pkt_rx2(:,10);

%%
%%

nbr1=max(pkt_rx1(:,1));nbr2=max(pkt_rx1(:,3));nbr3=max(pkt_rx1(:,5));
nbr4=max(pkt_rx1(:,7));nbr5=max(pkt_rx1(:,9));nbr6=max(pkt_rx2(:,1));
nbr7=max(pkt_rx2(:,3));nbr8=max(pkt_rx2(:,5));nbr9=max(pkt_rx2(:,7));
nbr10=max(pkt_rx2(:,9));

%%
%%

contador1=1; contador2=1; contador3=1; contador4=1; contador5=1; contador6=1; contador7=1;



contador8=1; contador9=1; contador10=1;

for n= 1:nbr1

PER1=1-(Sim1(n)/1000); M1(contador1)=PER1; format long;
BER1=1-nthroot (1-PER1,1000); SNR11= SNR1(n); M11(contador1)=SNR11;
contador1=contador1+1;
end

for n= 1:nbr2

PER2=1-(Sim2(n)/1000); M2(contador2)=PER2; format long;
BER2=1-nthroot (1-PER2,1000); SNR22= SNR2(n); M22(contador2)=SNR22;
contador2=contador2+1;
end

for n= 1:nbr3

PER3=1-(Sim3(n)/1000); M3(contador3)=PER3; format long;
BER3=1-nthroot (1-PER3,1000); SNR33= SNR3(n); M33(contador3)=SNR33;
contador3=contador3+1;
end

for n= 1:nbr4

PER4=1-(Sim4(n)/1000); M4(contador4)=PER4; format long;
BER4=1-nthroot (1-PER4,1000); SNR44= SNR4(n); M44(contador4)=SNR44;
contador4=contador4+1;
end

for n= 1:nbr5

PER5=1-(Sim5(n)/1000); M5(contador5)=PER5; format long;
BER5=1-nthroot (1-PER5,1000); SNR55= SNR5(n); M55(contador5)=SNR55;
contador5=contador5+1;
end

for n= 1:nbr6

PER6=1-(Sim6(n)/1000); M6(contador6)=PER6; format long;
BER6=1-nthroot (1-PER6,1000); SNR66= SNR6(n); M66(contador6)=SNR66;
contador6=contador6+1;
end

for n= 1:nbr7

PER7=1-(Sim7(n)/1000); M7(contador7)=PER7; format long;
BER7=1-nthroot (1-PER7,1000); SNR77= SNR7(n); M77(contador7)=SNR77;
contador7=contador7+1;
end

for n= 1:nbr8

PER8=1-(Sim8(n)/1000); M8(contador8)=PER8; format long;



```
BER8=1-nthroot (1-PER8,1000); SNR88= SNR8(n); M88(contador8)=SNR88;
contador8=contador8+1;
end
```

```
for n= 1:nbr9
PER9=1-(Sim9(n)/1000); M9(contador9)=PER9; format long;
BER9=1-nthroot (1-PER9,1000); SNR99= SNR9(n); M99(contador9)=SNR99;
contador9=contador9+1;
end
```

```
for n= 1:nbr10
PER10=1-(Sim10(n)/1000); M10(contador10)=PER10; format long;
BER10=1-nthroot (1-PER10,1000); SNR101= SNR10(n); M101(contador10)=SNR101;
contador10=contador10+1;
end
```

%%%

```
snr1=mean(M11); snr2=mean(M22); snr3=mean(M33); snr4=mean(M44); snr5=mean(M55);
snr6=mean(M66); snr7=mean(M77); snr8=mean(M88); snr9=mean(M99); snr101=mean(M101);
```

%%%

```
SNR=[snr1,snr2,snr3,snr4,snr5]'; BER=[BER1,BER2,BER3,BER4,BER5]';
snr=[snr6,snr7,snr8,snr9,snr101]'; ber=[BER6,BER7,BER8,BER9,BER10]';
```

%%%

```
figure(1);semilogy(SNR,BER,'r');grid on
hold on
semilogy(snr,ber,'b');grid on
hold off
```