



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Pruebas de transmisión del estándar IEEE802.11 en Radio Definido por Software

TESINA

QUE PARA OBTENER EL TÍTULO DE

MAESTRO EN INGENIERÍA

PRESENTA

GIBRAN JOEL RESENDIZ TOLENTINO

DIRECTOR DEL TRABAJO:

DR. JAVIER GÓMEZ CASTELLANOS

Ciudad de México, Septiembre, 2019

Índice general

1. RESUMEN	1
1.1. Antecedentes	1
1.2. Radio Definido por Software	3
1.2.1. GNU Radio	3
1.2.2. UHD (USRP Hardware Driver)	3
1.3. Estado del Arte	3
1.4. Conceptos del Protocolo IEEE 802.11 “Wifi”	4
1.4.1. OFDM (Orthogonal Frequency Division Multiplexing)	5
1.4.2. FHSS (Frequency Hopping Spread Spectrum)	9
1.4.3. PLCP (Physical Layer Convergence Procedure)	10
2. OBJETIVOS	11
2.1. Propuesta	11
2.2. Objetivo General	11
2.3. Objetivos Específicos	12
3. IMPLEMENTACIÓN	13
3.1. Bloques en GNU Radio	14
4. METODOLOGÍA	16
4.1. Implementación del Pase de Mensajes	16
4.2. Seguimiento del Bloque 802.11 en GNU Radio	16

5. PRUEBAS DE VERIFICACIÓN	19
5.1. Primera Simulación	19
5.2. Segunda Simulación	21
6. TOPOLOGÍA DEL EXPERIMENTO	22
7. EXPERIMENTO DE VARIACIÓN EN MODULACIÓN	27
7.1. Procedimiento	27
7.2. Resultados	31
7.3. Análisis	32
8. EXPERIMENTO DE VARIACIÓN EN EL ANCHO DE BANDA	35
8.1. Procedimiento	35
8.2. Resultados	36
8.3. Análisis	36
9. EXPERIMENTO DE VARIACIÓN EN LA DISTANCIA	38
9.1. Procedimiento	38
9.2. Resultados	39
9.3. Análisis	41
10. EXPERIMENTO DE VARIACIÓN DE TASA DE TRANSMISIÓN	43
10.1. Procedimiento	43
10.2. Resultados	44
10.3. Análisis	46
11. CONCLUSIONES	47
12. TRABAJO FUTURO	50

Índice de cuadros

6-1. Características de transmisión de USRP 2932.	23
6-2. Características de recepción de USRP 2932.	24
7-1. Porcentaje promedio de paquetes recibidos en diferentes modulaciones.	33
10-1. Comparación de throughput entre métricas.	46

Índice de figuras

1-1. Estructura de USRP.	2
1-2. Representación del sistema OFDM.	6
1-3. Arquitectura OFDM.	7
4-1. Diagrama de bloques implementado en GNU Radio para 802.11 Wifi Receptor. .	17
5-1. Captura de paquetes de un AP mediante NIC de laptop con Wireshark.	20
5-2. Captura de paquetes mediante USRP	20
5-3. Comunicación entre USRP's.	21
6-1. Conexión entre USRP's.	23
6-2. Bloque de Intervalo entre mensajes.	24
6-3. Captura de datos mediante USRP, GNU Radio y Wireshark.	25
6-4. Monitoreo de características de transmisión en Iperf.	26
6-5. Monitoreo de características de transmisión en Iperf.	26
7-1. Interfaz de Recepción.	28
7-2. Interfaz de Transmisión.	28
7-3. Constelaciones GNU Radio.	29
7-4. Gráfica de paquetes recibidos variando la modulación.	30
7-5. Dispersión en constelación de recepción.	30
7-6. Comparación entre AP y USRP con modulación 64QAM. a) Constelación por USRP, b) constelación por AP.	31

7-7. Porcentaje de paquetes recibidos exitosamente con variación de modulación. . .	32
7-8. 20 Porcentaje de paquetes recibidos exitosamente para todas las modulaciones . .	34
8-1. Porcentaje de paquetes recibidos exitosamente con ancho de banda de 5 [MHz . .	36
8-2. Recepción de paquetes malformados.	37
9-1. Porcentaje de paquetes recibidos exitosamente con variación de distancia.	39
9-2. Gráfica de cantidad de paquetes recibidos desde un AP con variación de distancia. 40	
9-3. Constelación de recepción de AP con ancho de banda de 20 [MHz	40
10-1. Gráfica de cantidad de paquetes recibidos con variación de intervalo entre pa- quetes antes de 100 ms	44
10-2. Gráfica de cantidad de paquetes transmitidos con variación de intervalo entre paquetes después de 100 ms	45
10-3. Porcentaje de paquetes recibidos exitosamente con variación en Delay	45

Capítulo 1

RESUMEN

El presente trabajo implementa el estándar IEEE 802.11, “Wifi”, con la versión desarrollada en [1], mediante una plataforma de soporte SDR (Radio Definido por Software), cuya función es facilitar el acceso al hardware de dispositivos utilizados para radiocomunicaciones mediante una programación enfocada a conexión entre bloques. Para realizar la implementación primero se buscará el contexto actual de los trabajos relacionados, después se expondrán las herramientas utilizadas para la realización, tanto en una plataforma digital como en hardware. Se explicarán aspectos de relevancia con las pruebas que se realicen. Se desglosará el procedimiento de las pruebas de simulación, así como en la obtención de resultados. Por último, se hará un análisis de las pruebas y sus resultados.

1.1. Antecedentes

El concepto SDR se define en [2] como un conjunto de DSP's (Digital Signal Processing) primitivos, un sistema de meta niveles que combinan funciones de los sistemas de comunicación y un conjunto de procesadores objetivo que permiten que la comunicación sea implementada en tiempo real, además de disponer de una etapa de radiofrecuencia. Para adecuarse al sistema de comunicación que se requiere implementar se reconfigura el radio de radiofrecuencia SDR a través del software. Existen dos plataformas en las que se puede implementar SDR (GNU Radio

y Labview); estas cuentan con un entorno gráfico que facilita su manejo. Lenguajes como C++, Python y MATLAB también se utilizan para el desarrollo, de hecho, estas dos plataformas se encuentran programadas en los lenguajes mencionados. En este trabajo se utilizó la plataforma GNU Radio por su fácil interacción con el código, además de la tendencia actual hacia el código libre que vienen ofreciendo las nuevas tecnologías. El dispositivo que se utiliza en el presente trabajo, que interactúa con GNU Radio, es identificado como USRP 2932, por sus siglas en inglés (Universal Software Radio Peripheral). A continuación, se muestra en Fig. 1-1 un diagrama donde se encuentra del lado izquierdo la etapa de procesamiento en banda base y del lado derecho la etapa de tratamiento de radiofrecuencia de las señales en un USRP. Riset en [3], clasifica el manejo de GNU Radio representado en Fig. 1-1; la sección izquierda del diagrama siendo el back-end (procesamiento digital), mientras que la sección derecha representa el front-end (circuitos de radiofrecuencia).

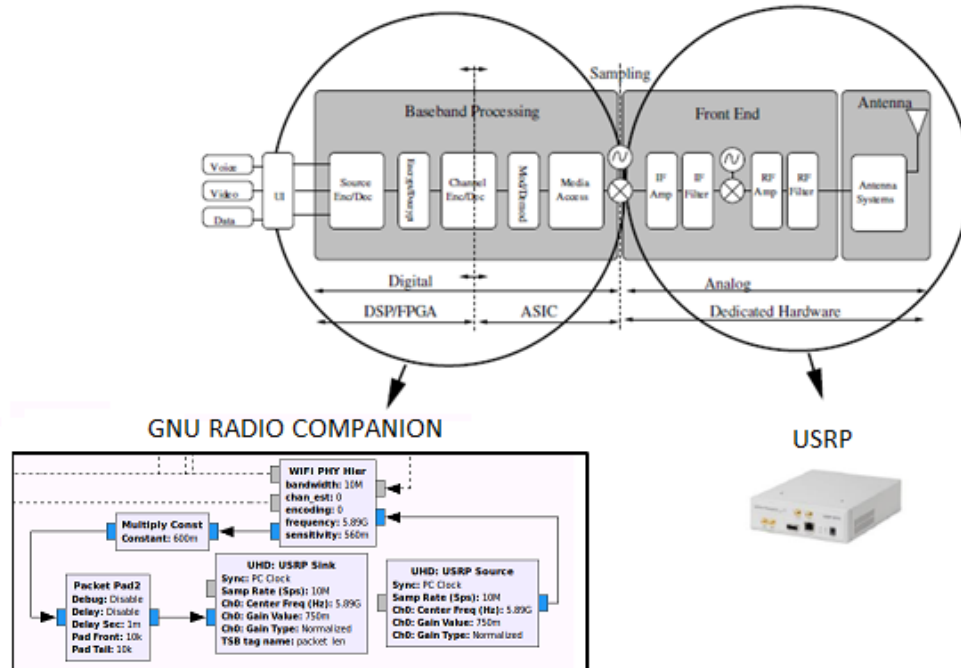


Fig. 1-1. Estructura de USRP. Riset T. (2016). Introduction to GNU Radio. [Figura]. Recuperado de <https://gnuradio-fr-18.sciencesconf.org>

1.2. Radio Definido por Software

1.2.1. GNU Radio

La definición de GNU Radio en [4] es descrita como un software de desarrollo libre y de código abierto, éste implementa software que manipula las señales de radiofrecuencia. Puede ser utilizado con dispositivos externos con hardware de radiofrecuencia, o sin hardware en una simulación con el ambiente. Utiliza una plataforma digital que facilita su programación y lo hace de manera interactiva, dicha plataforma se llama GRC (GNU Radio Companion). GNU Radio tiene dos conceptos importantes que son flujo de grafos y bloques, el primero tiene mayor representación en lo que corresponde la Teoría de Grafos y por este medio fluyen los datos de bloque a bloque. Los bloques representan los llamados nodos en Teoría de Grafos y la conexión entre bloques correspondería a las ramas.

1.2.2. UHD (USRP Hardware Driver)

UHD diseñada por Ettus USRP ¹, es básicamente una API propia de los dispositivos SDR donde se puede programar hardware mediante una plataforma digital, así como algunas partes analógicas. El desarrollo de esta herramienta se puede hacer tanto en lenguaje de programación Python como en C. Una característica a considerar en el manejo de SDR con GRC es la compatibilidad que tiene la versión de GNU Radio instalada con el tipo de USRP o SDR utilizado. Al igual que GNU Radio, la instalación puede ser desde fuente o binaria, además de tener facilidades en la manipulación del FPGA.

1.3. Estado del Arte

Es importante revisar los recientes trabajos relacionados, así como aquellos conceptos referentes al entorno del software ocupado para los dispositivos USRP (Universal Software Radio Peripheral), siendo fundamental para llevar acabo la simulación del algoritmo desarrollado por Chia-Horn en [5] donde implementa el estándar IEEE 802.11. La primera implementación del

¹<http://www.ettus.com/>

estándar fue hecha en [6] por la Universidad de Salento, en donde implementaron la parte de transmisión, posteriormente Bloessl en conjunto con sus colegas detalla en [7] la construcción del receptor en GNU Radio. Por otro lado, existen trabajos que fueron implementados en LabVIEW, estos tienen un entorno distinto, sin embargo, al igual que GNU Radio maneja el flujo de bloques.

1.4. Conceptos del Protocolo IEEE 802.11 “Wifi”

De las partes fundamentales del estándar IEEE 802.11 con la versión estandarizada en 1999, rescatada de [1], OFDM (Orthogonal Frequency Division Multiplexing) es un esquema de multiplexación de frecuencia de gran relevancia, OFDM es la mejora del sistema FDM (Frequency Division Multiplexing), de la misma manera, OFDM mejora la comunicación en el estándar IEEE 802.11. Como parte esencial en transmisión de tecnologías inalámbricas el uso de OFDM es primordial en la implementación. La recopilación de trabajos para su desarrollo en GNU Radio se ha trabajado en la capa física del modelo OSI (Open Systems Interconnection), estandarizado por la organización ISO (International Organization for Standardization). Por lo que se expondrá al lector lo más cercano posible a una descripción del estándar IEEE 802.11, evitando las normas y características rigurosas y con la omisión de técnicas, algoritmos o simulaciones poco relevantes para el desarrollo del reporte, tanto de la capa física como de subsecuentes capas del modelo OSI. En el presente trabajo se dará por hecho que el diseño del algoritmo OFDM en [5], la construcción del transmisor IEEE 802.11 en [6] y la construcción del receptor IEEE 802.11 en [7], además de las herramientas rescatadas del estado del arte, colaboran con la capa física y la de enlace de datos del modelo OSI sin problemas de compatibilidad o la forma en que opera el estándar IEEE 802.11. Dicho lo anterior, se pretende facilitar la comprensión del contexto y el desarrollo como tal.

1.4.1. OFDM (Orthogonal Frequency Division Multiplexing)

De acuerdo con Chia-Horng en su trabajo “On the design of OFDM signal detection algorithms for hardware implementation”, OFDM es conocida como una tecnología de acceso al medio, o también, como una técnica de transmisión de portadoras [8]. Lo cierto es que OFDM trabaja con sistemas de comunicación cuyas tasas de transmisión son altas y se concentra en el transporte de multiportadoras. Su principio de operación se basa en el envío de portadoras a través del manejo de espectro (Spread Spectrum ²). Dichas portadoras tienen como característica la ortogonalidad entre sí mismas en el dominio de la frecuencia. Para lograr esta ortogonalidad, el mecanismo OFDM ocupa la herramienta matemática conocida como FFT (Fast Fourier Transform), haciendo factible la manipulación de la señal en el espectro de frecuencias en forma discreta. Con la ortogonalidad de portadoras en el esquema OFDM, al comparar con otros sistemas de distribución de internet, tanto en redes celulares como Wifi, el ahorro del ancho de banda es mayor y a su vez la capacidad de datos que se pueden transmitir. Ergen en [8] menciona que el mecanismo del sistema OFDM tiene como objetivo pasar de un flujo de datos serial a uno paralelo, después de ello se asocia una cantidad de bits ordenada a un número complejo, para ser transmitidos dentro de un sistema PSK o QAM. Todo dentro de un flujo de datos que se encuentra asociado a una portadora. A continuación, en Fig. 1-2 se muestra el sistema básico OFDM descrito en el párrafo anterior, el diagrama es recuperado de [8], donde Ergen explica con mayor detalle el funcionamiento de cada bloque.

²El sistema Spread Spectrum es un sistema, el cual, la señal transmitida sobre un ancho de banda en frecuencia es mucho más grande que el mínimo ancho de banda requerido para transmitir la información que se envía. [9]

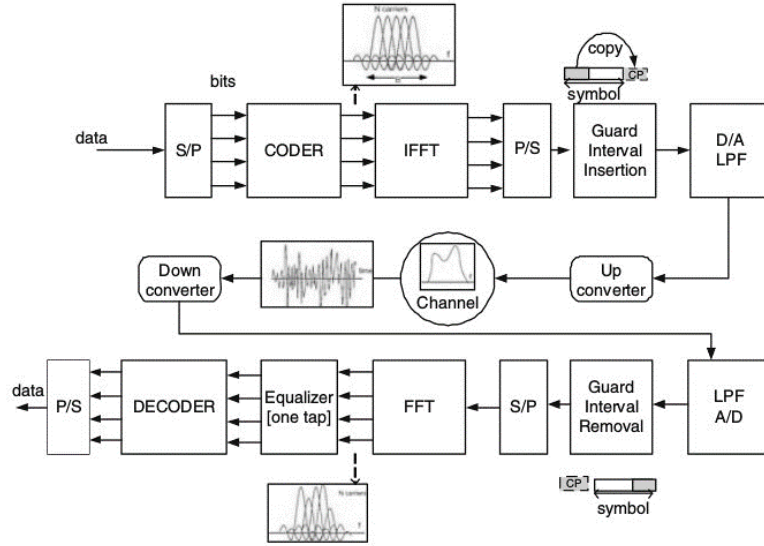


Fig. 1-2. Representación del sistema OFDM. Copyright 2009 por M. Erge. Reimpresión con permiso. Recuperado de [8]

El esquema de los procesos más relevantes de una arquitectura 802.11 se observa en Fig. 1-3, cada proceso se lista a continuación con una descripción resumida de "Mobile Broadband Including WiMAX and LTE" escrita Ergen en [8] y el trabajo desarrollado por Henríquez en [18].

- *Coding (Codificación de Canal)*: Este proceso es útil para corregir un cierto número de errores en tasas de transmisión altas donde las señales de transmisión son más propensas a desvanecimientos e interferencias. Junto con la etapa de interleaving es parte de lo que se denomina codificación de canal, consta de dos subcapas, la primera es llamada *Randomizer* y sirve para conseguir una densidad de potencia más uniforme en el ancho de banda de transmisión para cualquier tipo de portadora digital transmitida mediante la aleatorización de los datos, la segunda etapa es llamada *FEC (Forward Error Correction)* y sirve para corregir los datos erróneos en recepción mediante bits de redundancia, está conformado por la concatenación de un código *Reed-Solomon* y una tasa compatible con un código convolucional.
- *Interleaving (Entrelazado)*: Esta etapa separa la información de forma pseudo-aleatoria

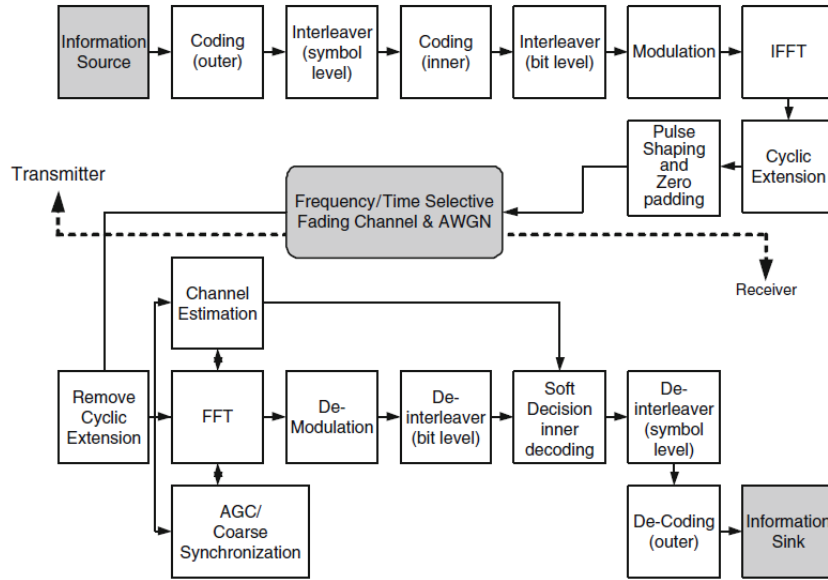


Fig. 1-3. Arquitectura OFDM. Copyright 2009 por M. Erge. Reimpresión con permiso. Recuperado de [8]

de tal manera que cuando la información de una cadena de bits pase por una ráfaga de error, los errores serán independientes, puede ser tanto en tiempo como en frecuencia.

- *Modulation (Modulación)*: En esta etapa se le asigna el tipo de modulación que transportará a la señal, en el estándar IEEE 802.11. Las modulaciones utilizadas son BPSK, QPSK, 16QAM y 64QAM.
- *FFT (Transformada Rápida de Fourier)*: Con esta técnica matemática se reduce la complejidad en la recepción de señal debido a que las operaciones del símbolo OFDM son del orden $N \log(N)$. Para la transmisión se utiliza la Transformada Inversa Rápida de Fourier, además, estas operaciones se pueden utilizar tanto en frecuencia como en tiempo.
- *Cyclic Extension (Inserción del prefijo cíclico)*: Busca evitar la Interferencia Intersimbólica que pueden causar los retardos de propagación y las multitrayectorias. Las N señales ortogonales divididas por el proceso OFDM son representadas por símbolos en la recepción, para que las señales no interfieran entre sí, se le añade un periodo de guarda conocido como prefijo cíclico.

- *Pulse Shaping (Muestreo de pulsos)*: Esta etapa es usada para espaciar el espectro, normalmente se utiliza con ventanas de tiempo de cosenos o pasa a través de filtros. Es la última etapa y hace referencia a la propia transmisión digital.

A continuación, se mencionarán los pasos importantes del protocolo IEEE 802.11 de acuerdo con Bloessl en [7] y Abrahantes en [9], que así mismo son las partes que ayudaron a construir el receptor Wifi en GNU Radio representado en Fig 4-1:

- **Detección de la Trama**: Esta etapa está compuesta de los primeros 8 bloques del diagrama de recepción, con las muestras obtenidas de los radios USRP se trata para convertir los datos de complejos a flotantes, son necesarios para la siguiente etapa de decodificación de la trama. Su desarrollo puede ser mejor explicado en el capítulo 4.2
- **Corrección de Desplazamiento de Frecuencia**: Es la transición del dominio del tiempo al dominio de la frecuencia, su bloque representativo es el bloque **WiFi Sync Long**. En ella se encuentra la Corrección de la Frecuencia de Offset y La alineación de los símbolos. La Corrección de Frecuencia de Offset es la encargada de representar el oscilador local y con ello la manipulación de frecuencias.
- **Alineación de Símbolos**: Es representada por el bloque **Wifi Sync Short**. El símbolo OFDM tiene 80 muestras, consiste de 16 muestras del prefijo cíclico y 64 de datos. La tarea de la alineación del símbolo es calcular dónde el símbolo empieza para extraer el símbolo de datos y diseñarlos para aplicar la FFT. Esta alineación es hecha con la ayuda de una secuencia de entrenamiento larga de 64 muestras de patrones largos
- **Corrección de Desplazamiento de Fase**: Este proceso puede verse reflejado en el bloque FFT encargado de la transición del dominio de frecuencia al dominio del tiempo. Para que el transmisor y el receptor se sincronicen se debe agregar un corrimiento de fase, este puede ser corregido con ayuda de portadoras pilotos y su bloque es representado por **WiFi Frame Equalizer**.

- Estimación de Canal: Es representado por el bloque **WiFi Frame Equalizer**, el cual también corrige la magnitud de la portadora eliminando a las subportadoras piloto y de guarda, así como la de DC
- Decodificación de Campo de Señal: Este proceso se encuentra en el bloque **OFDM Decode Signal**. Tiene un campo denominado SIGNAL, el cual acota el flujo de muestras con una etiqueta, transportando una tupla de codificación y la longitud del frame.
- Decodificación de la Trama: Este proceso se encuentra en el bloque **WiFi Decode MAC** y su función es decodificar la carga útil. Recibe los 48 portadoras de datos por símbolos OFDM que se traducen en el proceso de bloque en bloque a 48 vectores que con valores complejos

1.4.2. FHSS (Frequency Hopping Spread Spectrum)

En el desarrollo de [5], [7] y la interfaz GRC, el diagrama de bloques del estándar IEEE 802.11 con la versión del año 1999 [1], ocupan la técnica FHSS (Frequency Hopping Spread Spectrum) para la estructura de la capa física del modelo OSI, siendo una técnica de transmisión del protocolo Wifi cuyas dos principales funciones de acuerdo al estándar IEEE 802.11 en el apartado 14 son:

- Función de convergencia en la capa física. Adapta el sistema PMD (Physical Medium Dependent) a los servicios que ofrece la capa física. Esta función es soportada por PLCP (Physical Layer Convergence Procedure), la cual define un método de mapeo de las unidades de datos de la subcapa MAC (Medium Access Control) dentro de un formato de trama para enviar y recibir datos de usuarios, además de manejar información entre dos o más STA's ³.
- PMD Physical Medium Dependent. El método de transmisión y recepción de datos mediante un medio inalámbrico entre dos o más STA's.

³STA se define en [1] como algún dispositivo que contenga una conformación del protocolo IEEE 802.11.

1.4.3. PLCP (Physical Layer Convergence Procedure)

PLCP es fundamental en [5], las dos funciones descritas anteriormente conforman la subcapa PLCP, de la misma manera conforma las dos divisiones fundamentales en los bloques de GNU Radio.

Capítulo 2

OBJETIVOS

Con el crecimiento de tecnologías referentes a dispositivos de IoT ⁴ se incrementa la necesidad de diseñar redes y algoritmos que sean escalables. Para evaluar la escalabilidad y robustez de dichas redes es necesario generar pruebas en diversas métricas como la tasa de transmisión, la modulación, la distancia entre dispositivos y el ancho de banda.

2.1. Propuesta

Se realizará la implementación del protocolo IEEE 802.11 mediante sólo un transmisor y un receptor utilizando dos USRP's 2932 y GNU Radio. Al hacer la simulación de WiFi se espera una comprensión del funcionamiento de una transmisión en un ambiente de SDR.

2.2. Objetivo General

Tomando el trabajo desarrollado por Boessl en [7], se implementará el estándar IEEE 802.11 Wifi en radios definidos por software (USRP N2932) mediante la plataforma de programación GNU Radio. Después, para cuatro experimentos, se hará un análisis de resultados de la capacidad efectiva de transferencia de datos obtenidos durante el enlace de cada experimento. Los

⁴El concepto de Internet de las cosas (IoT) busca la difusión del Internet, siendo una extensión de este, transfiriendo contenido a dispositivos de diferentes aplicaciones. [10]

experimentos variarán la tasa de transmisión, la modulación, la distancia entre dispositivos y el ancho de banda.

2.3. Objetivos Específicos

- Desarrollo del protocolo Wifi en SDR.
- Comprensión del funcionamiento de Wifi
- Simulaciones del estándar IEEE 802.11
- Manejo de la plataforma GNU Radio

Capítulo 3

IMPLEMENTACIÓN

Se generó la instalación de GNU Radio en dos computadoras diferentes con el mismo sistema operativo (Ubuntu 16.04), en una se generó la instalación desde fuente y en otra de forma binaria. La instalación desde fuente es la construcción de GNU Radio y la instalación de todos los códigos utilizados que generan los bloques y hacen funcional el procesamiento del programa. La instalación binaria es la misma instalación, pero con códigos binarios, con excepción de los códigos que generan la compilación de GRC. Entonces, se lista a continuación las herramientas utilizadas para la simulación:

- USRP 2932
- Antenas VERT2450
- Cabe Ethernet 4a categoría
- Sistema Operativo Ubuntu 16.04
- Adaptador de red 10Gigabit Ethernet
- Analizador de red Wireshark 2.6

Se conectan dos USRP's en dos computadoras diferentes, si éstas no cuentan con conexión 10Gigabit Ethernet necesitan de un adaptador de red 10Gigabit Ethernet. Se instalan las antenas

antes de prender los radios, se actualiza el software que soporta GNU Radio y LabVIEW, con estos pasos anteriores se puede proceder a programar en el software de preferencia.

3.1. Bloques en GNU Radio

La composición de GNU Radio es un conjunto de herramientas de programación que desarrollan una variedad de protocolos de telecomunicaciones, interactúa directamente con el hardware que ocupan diversos sistemas de comunicación a través de la estructura de bloques programados. GNU Radio puede resumirse en conceptos fundamentales; el tipo de dato, la conexión entre bloques y los puertos de entrada y salida. En la siguiente parte del Código 1 se muestra la construcción de un bloque en GNU Radio, con el fin de ejemplificar las características básicas de los bloques ya implementados en la interfaz GRC. La línea 1 define la función de mayor jerarquía, la línea 2 define la variable a utilizar, la línea 3 define el tipo de bloque y su construcción, la línea 4 define el nombre del atributo del bloque y las líneas 5 y 6 definen el tipo de dato utilizado tanto en el puerto de entrada como en el de salida.

```
1     def __init__(self,schedule):
2         self.schedule=schedule
3         gr.sync_block.__init__(self,
4             name="sleep",
5             in_sig=[numpy.complex64],
6             out_sig=[numpy.complex64])
```

Código 1.

La programación es realizada en lenguaje Python, a pesar de ello, es necesario programar un conjunto de códigos escritos en Python, C++ y XML para la generación de la interfaz gráfica y la compatibilidad con otros bloques. Teniendo más clara la estructura de GNU Radio se puede describir cómo construir los bloques en los que se trabaja y que no se encuentran instalados en GNU Radio, así como son los bloques diseñados por Bloessl en [7]. La secuencia de comandos en Código 2 genera la construcción del bloque “ieee802.11” al escribirlos en consola con sistema operativo Ubuntu de cualquier distribución.

```
1 git clone git://github.com/bastibl/gr-ieee802-11.git
2 cd gr-ieee802-11
3 mkdir build
4 cd build
5 cmake ..
6 make
7 sudo make install
8 sudo ldconfig
```

Código 2.

Existe un bloque en GNU Radio llamado “foo”, el cual, hace la captura de paquetes en archivos pcap, debido a las complicaciones en su construcción se descarga directamente desde la página web recuperada de [<https://github.com/bastibl/gr-foo>]. Además de ello, es necesario leer y seguir los pasos descritos en el archivo README adjunto en las carpetas de la biblioteca “ieee802.11”. Cabe aclarar, que no es requerido hacer una instalación de GNU Radio desde fuente, sin embargo, puede generar una mejor comprensión de cómo se programan los bloques y sus características. Además, su instalación de fuente permite modificar el código de los bloques. De esta manera, los bloques dedicados para el estándar IEEE 802.11 serán más fáciles de comprender para su desarrollador.

Capítulo 4

METODOLOGÍA

4.1. Implementación del Pase de Mensajes

Como ya se había comentado GNU Radio fue diseñado con el mecanismo del pase de datos entre bloques, sin embargo, su desempeño en el pase de mensajes, metadatos o estructura de paquetes era escaso hasta la introducción del concepto “Tag Stream”. La herramienta Tag Stream en GNU Radio se utiliza en paralelo al flujo de datos PMT ⁵. Al trabajar en paralelo estructuras como el pase de mensajes necesarias en redes de datos facilitan el desarrollo de protocolos con conexión a Internet.

4.2. Seguimiento del Bloque 802.11 en GNU Radio

Las características implementadas en GNU Radio están limitadas a las capacidades del mismo software. A continuación, en Fig. 4-1, se presentará el diagrama de bloques básico y de configuración libre proveído por [5], diseñado para la recepción del protocolo IEEE 802.11. Por [7] se entiende que el script diseñado en la interfaz GRC tiene dos tareas principales; la detección de la trama y la decodificación de la trama.

⁵Son datos cuya estructura de datos no está definida y son diseñados como un contenedor genérico de datos que hace seguro el traslado de información entre bloques e hilos en GNU Radio. [11]

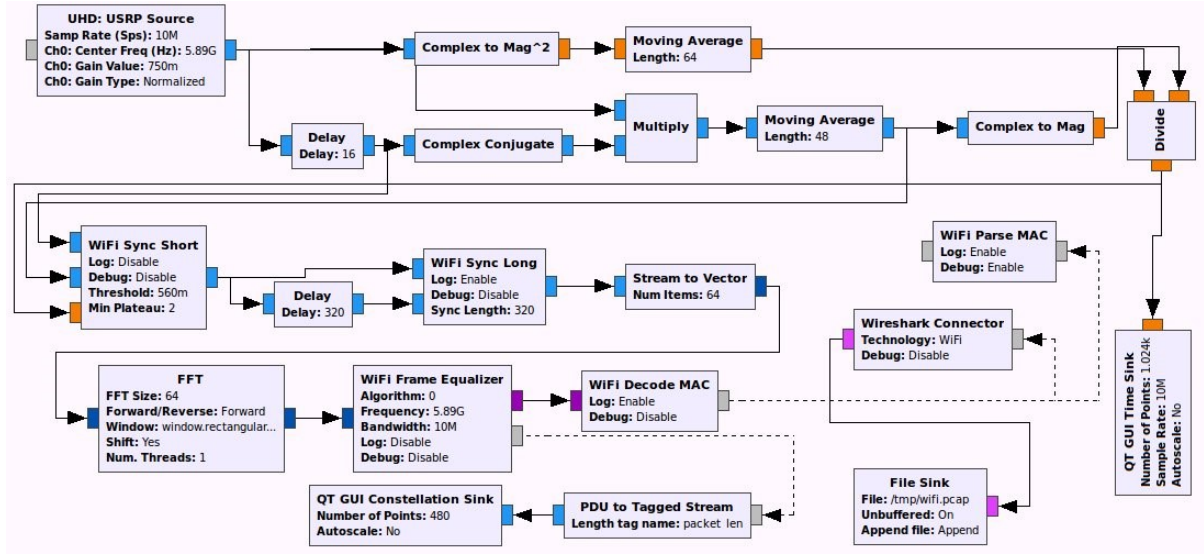


Fig. 4-1. Diagrama de bloques implementado en GNU Radio para 802.11 Wifi Receptor. B. Bloessl 2013. [Diagrama]. Recuperado de [7].

Para tener el precepto de la construcción en GNU Radio del protocolo 802.11 se describen los bloques relevantes de la interfaz. Más aún, se pretende dar un panorama del esquema que especifique cómo el diagrama de bloques implementado en GRC incluye las partes primordiales del desarrollo de Wifi. La construcción de los primeros 8 bloques del diagrama de recepción en Fig. 4-1 está basada en el algoritmo de detección de la trama en [5], su funcionamiento, en breves palabras, se describe en [7] como el manejo del valor de autocorrelación al ingreso del muestreo en el flujo de datos. Su ecuación se muestra en (4-1).

$$a[n] = \sum_{k=0}^{N_{win}-1} s[n+k] \bar{s}[m+k+16] \quad (4-1)$$

Donde n es la cantidad de muestras recuperadas en la recepción, s es la función de muestreo de la señal recibida, \bar{s} denota el complejo conjugado de s , k aumenta la suma de los coeficientes para el valor de autocorrelación, N_{win} es la ventana ajustable que actúa como un filtro paso-bajas y el factor 16 es el intervalo entre las muestras al aplicar la FFT. La ecuación en (4-1) representa la acumulación de portadoras, actuando el factor 16 como un factor de corrimiento para un filtro paso-bajas. La representación de la ecuación (4-1) en el diagrama de Fig. 4-1 se

basa en el preámbulo PLCP y a su vez en un conjunto de números complejos representados tal como lo hace en [5]. Los primeros ocho bloques de Fig. 4-1 son parte de la detección de la trama, los bloques restantes son la decodificación de la trama reportado por Bloessl en [7]. Cabe destacar que, al tener una conexión entre transmisor y receptor, en cuanto a la parte analógica, la implementación es más simple. Para la parte de la transmisión del protocolo IEEE 802.11 en GNU Radio, se tomaron las herramientas ya implementadas en las paqueterías de GNU Radio, tal como lo es la arquitectura de OFDM. Se hace una división en tres principales bloques que componen el esquema por el lado de la transmisión; FFT Shifting, Carrier Indexing y Carrier and Symbol Allocation. Con estos "bloques" se genera una transmisión aproximada a lo que es el estándar 802.11. Al trabajar con un protocolo de redes de datos inalámbricas, es un hecho que sus restricciones se encuentran en la cantidad de información transmitida, ancho de banda y tiempos de transmisión, además de la tasa de transferencia de datos del hardware utilizado. Dadas las restricciones, se realizarán experimentos que verifiquen las limitantes y a su vez expongan las ventajas.

Capítulo 5

PRUEBAS DE VERIFICACIÓN

Para verificar el funcionamiento de la transmisión se realizaron dos pruebas que determinen la operabilidad de una transmisión común con el estándar IEEE 802.11. La primera prueba pretende comparar la captura de paquetes mediante una tarjeta de red de una laptop contra la captura de paquetes del radio USRP 2932 y la construcción descrita en el capítulo 4. La segunda prueba tiene como finalidad comparar los paquetes transmitidos y recibidos entre dos radios USRP 2932 con la misma construcción del módulo `ieee80211` en la interfaz GRC.

5.1. Primera Simulación

Las primeras simulaciones se muestran en Fig. 5-1 y 5-2, Fig. 5-1 muestra la captura de paquetes Wifi de un AP ⁶ (Access Point) durante 140 segundos mediante la NIC de una Laptop con ayuda de Wireshark. Fig. 5-2 muestra la captura de paquetes mediante la recepción de USRP y GNU Radio con Wireshark en los mismos 140 segundos de captura mediante la NIC de la laptop. La captura de paquetes del conjunto GNU Radio y USRP es mayor que la captura de una interfaz de red común, con un pico de captura de 30 paquetes por segundo mientras que el pico de la interfaz de red de una laptop es de 20 paquetes por segundo, la diferencia puede ser causada por las conexiones de internet que no soporta al detectar el receptor USRP

⁶El AP utilizado fue un modem con características de 802.11 b/g/n a 2.4GHz velocidad de 300Mbps y MIMO 2T2R.

y el algoritmo de recepción utilizado de la ecuación (4-1). Las gráficas representan los paquetes capturados por cada segundo, los picos de las gráficas basicamente son la acumulación de paquetes por segundo capturados, en algunos casos se interpretar como la máxima ventana de contención de transmisión.

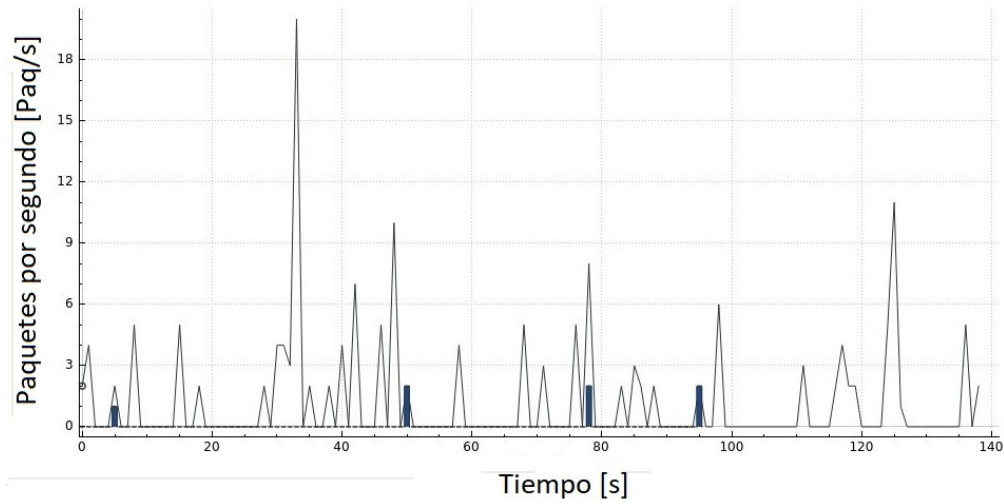


Fig. 5-1. Captura de paquetes de un AP mediante NIC de laptop con Wireshark.

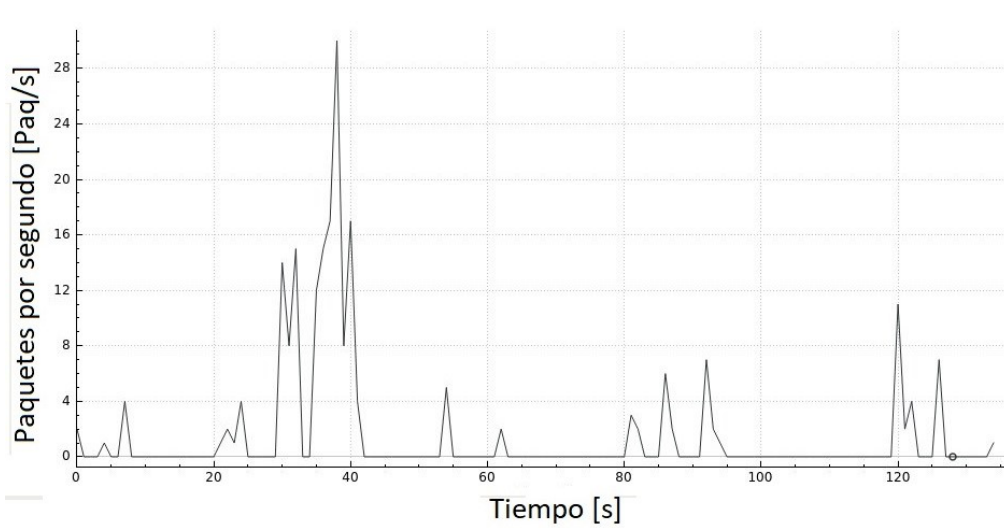


Fig. 5-2. Captura de paquetes mediante USRP.

5.2. Segunda Simulación

En Fig. 5-3 se muestra la comparación entre transmisión y recepción de paquetes con la plataforma GNU Radio.

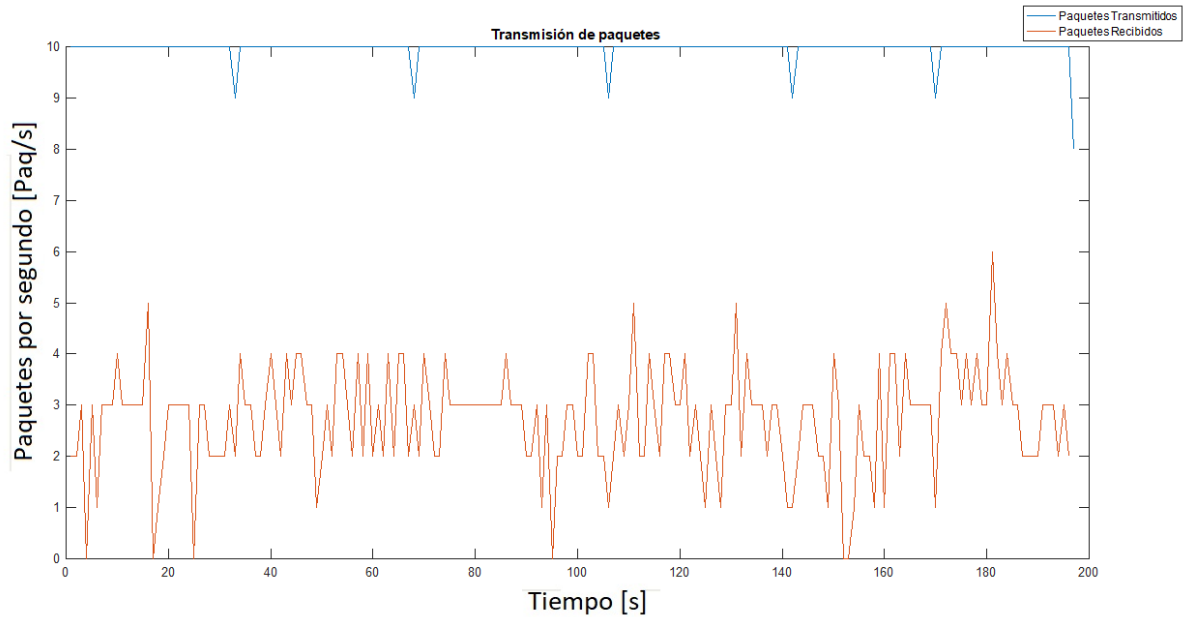


Fig. 5-3. Comunicación entre USRP's.

Los resultados en Fig. 5-3 muestran que menos del 50 % de paquetes son exitosamente recibidos. Las características utilizadas para esta segunda simulación de comunicación fueron las siguientes:

- Ancho de banda de la aplicación en GNU Radio 5 [MHz]
- Intervalo entre paquetes de 100 [ms]
- Modulación BPSK 1/2
- Tamaño del paquete 1500 [Bytes]
- Frecuencia de transmisión 4.4 [GHz] pensada para 5.89 [GHz]

Capítulo 6

TOPOLOGÍA DEL EXPERIMENTO

En los siguientes capítulos se realizarán cuatro experimentos; de variación en la modulación, variación en el ancho de banda, variación en la distancia y variación de tasa de transmisión. Para generar los experimentos se utilizarán en principio las mismas características utilizadas en la segunda simulación del capítulo anterior, sin embargo, se variará una de estas características dependiendo el experimento con excepción del experimento de variación en la distancia que sólo variará la distancia entre radios. Para el caso de la variación de modulación se variarán las modulaciones con seis opciones (BPSK 1/2, BPSK 3/4, QPSK 1/2, QPSK 3/4, 16QAM 1/2, 16QAM 3/4, 64QAM 1/2 64QAM 3/4). Para el caso de la variación en el ancho de banda la aplicación cuenta con tres opciones (5 MHz, 10 MHz y 20 MHz). Para el caso de la variación de distancia se variará la distancia entre radios hasta observar dispersión en la constelación. Para el último experimento de variación se generará la simulación variando el intervalo entre paquetes de 100 milisegundos hasta 1 segundo. El esquema de Fig. 6-1 representa cómo se llevarán a cabo los experimentos entre los radios USRP 2932 y las tablas 6-1 y 6-2 muestran las características de los mismos USRP's.



Fig. 6-1. Conexión entre USRP's.

Cuadro 6-1: Características de transmisión de USRP 2932.

Rango de Frecuencia	400 MHz a 4.4 GHz
Pasos de Frecuencia	<1 kHz
Potencia de salida máxima	50 mW a 100 mW (17 dBm a 20 dBm)
Rango de Ganancia	0 dB a 31 dB
Pasos de Ganancia	0.5 dB
Máximo Ancho de Banda Instantáneo en tiempo real	
Ancho de banda a muestra de 16 bits	20 MHz
Ancho de banda a muestra de 32 bits	40 MHz
Máxima tasa de muestreo I/Q	
Ancho de banda a muestra de 16 bits	20 MHz
Ancho de banda a muestra de 32 bits	40 MHz
DAC	2 canales, 400 MS/s, 16 bits
Rango Dinámico libre de Spurius	80 dB

Cuadro 6-2: Características de recepción de USRP 2932.

Rango de Frecuencia	400 MHz a 4.4 GHz
Pasos de Frecuencia	<1 kHz
Potencia de entrada máxima	0 dBm)
Rango de Ganancia	0 dB a 31.5 dB
Pasos de Ganancia	0.5 dB
Figura de Ruido	5 dB a 7 dB
Máximo Ancho de Banda Instantáneo en tiempo real	
Ancho de banda a muestra de 16 bits	20 MHz
Ancho de banda a muestra de 32 bits	40 MHz
Máxima tasa de muestreo I/Q	
Ancho de banda a muestra de 16 bits	20 MHz
Ancho de banda a muestra de 32 bits	40 MHz
DAC	2 canales, 400 MS/s, 16 bits
Rango Dinámico libre de Spurius	80 dB

Después de hacer las conexiones debidas y asignar una dirección de subred dentro del sistema operativo utilizado (dirección habitual 192.168.10.2), se verifican aspectos básicos dentro de las peticiones de la plataforma utilizada GRC (GNU Radio Companion), como una capa MAC virtual “TUN/TAP”, los datos a transmitir, la banda de frecuencia, la captura de paquetes, la asignación del puerto, entre otras. En Fig. 6-2 se muestra la configuración de los datos del lado del transmisor y en Fig. 6-4 se muestra la debida recepción de estos.

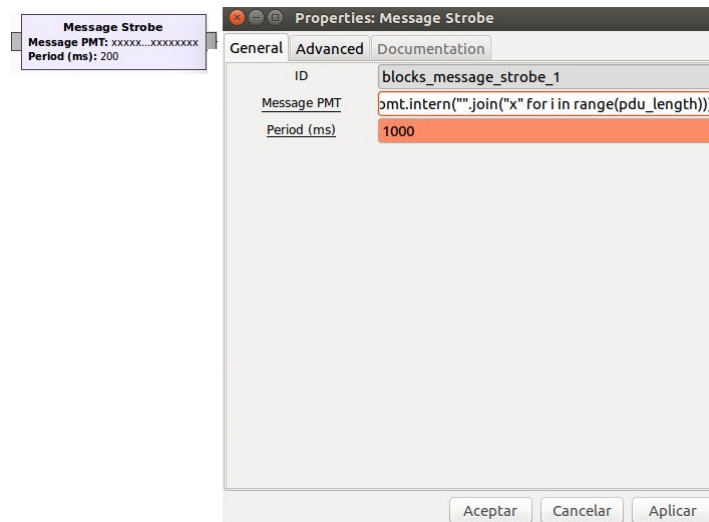


Fig. 6-2. Bloque de Intervalo entre mensajes.

En Fig. 6-2 se muestra el despliegue de variables del bloque “Message Strobe”. La funcionalidad de este bloque es el envío de mensajes a otro bloque cada cierto intervalo de tiempo, sin embargo, como la plataforma es amable al manejo del lenguaje Python, el comando que envía como mensaje es la iteración de la letra “x” convirtiéndose a un tipo de dato que se pueda trabajar en GRC llamado PMT y se muestra en Código 3.

```
1 pmt.intern("".join("x" for i in range(pdu_length)))
```

Código 3.

Existe una cabecera enfocada al protocolo IEEE 802.11 en GNU Radio llamada RFTap, la cual sólo es utilizada en SDR's y se encuentra estandarizada en Wireshark. Debido al hecho de no poseer algún tipo de encriptación, se puede acceder a la información contenida en los paquetes como se muestra en Fig. 6-4. Para la modificación del protocolo o las cabeceras, se modifica al código escrito en Python que es creado por la interfaz GRC, o simplemente se busca el protocolo de interés en la misma interfaz.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	23:23:23:23:23:23	42:42:42:42:42:42	LLC	1541	I, N(R)=60, N(S)=60; DSAP 0x78 Individual, SSAP 0x78 Command
2	1.076682	23:23:23:23:23:23	42:42:42:42:42:42	LLC	1541	I, N(R)=60, N(S)=60; DSAP 0x78 Individual, SSAP 0x78 Command
3	1.939822	23:23:23:23:23:23	42:42:42:42:42:42	LLC	1541	I, N(R)=60, N(S)=60; DSAP 0x78 Individual, SSAP 0x78 Command
4	2.421654	23:23:23:23:23:23	42:42:42:42:42:42	LLC	1541	I, N(R)=60, N(S)=60; DSAP 0x78 Individual, SSAP 0x78 Command
5	2.919240	23:23:23:23:23:23	42:42:42:42:42:42	LLC	1541	I, N(R)=60, N(S)=60; DSAP 0x78 Individual, SSAP 0x78 Command
6	3.388927	23:23:23:23:23:23	42:42:42:42:42:42	LLC	1541	I, N(R)=60, N(S)=60; DSAP 0x78 Individual, SSAP 0x78 Command
7	3.858848	23:23:23:23:23:23	42:42:42:42:42:42	LLC	1541	I, N(R)=60, N(S)=60; DSAP 0x78 Individual, SSAP 0x78 Command
8	4.336228	23:23:23:23:23:23	42:42:42:42:42:42	LLC	1541	I, N(R)=60, N(S)=60; DSAP 0x78 Individual, SSAP 0x78 Command
9	4.880293	23:23:23:23:23:23	42:42:42:42:42:42	LLC	1541	I, N(R)=60, N(S)=60; DSAP 0x78 Individual, SSAP 0x78 Command
10	6.286426	23:23:23:23:23:23	42:42:42:42:42:42	LLC	1541	I, N(R)=60, N(S)=60; DSAP 0x78 Individual, SSAP 0x78 Command
11	6.824769	23:23:23:23:23:23	42:42:42:42:42:42	LLC	1541	I, N(R)=60, N(S)=60; DSAP 0x78 Individual, SSAP 0x78 Command
12	7.515399	23:23:23:23:23:23	42:42:42:42:42:42	LLC	1541	I, N(R)=60, N(S)=60; DSAP 0x78 Individual, SSAP 0x78 Command

▼ Frame 1: 1541 bytes on wire (12328 bits), 1541 bytes captured (12328 bits)	
Encapsulation type: IEEE 802.11 plus radiotap radio header (23)	
Arrival Time: Feb 14, 2019 00:55:07.419537000 CST	
[Time shift for this packet: 0.000000000 seconds]	
Epoch Time: 1550127307.419537000 seconds	
[Time delta from previous captured frame: 0.000000000 seconds]	
[Time delta from previous displayed frame: 0.000000000 seconds]	
[Time since reference or first frame: 0.000000000 seconds]	
Frame Number: 1	
Frame Length: 1541 bytes (12328 bits)	
Capture Length: 1541 bytes (12328 bits)	
[Frame is marked: False]	
[Frame is ignored: False]	
Protocols in frame: radiotap>vlan>radiotap>llc>data1	
0000	00 00 11 00 0e 08 00 00 00 0c b2 00 00 00 18 00
0010	01 08 00 00 00 42 42 42 42 42 42 23 23 23 23 23
0020	23 ff ff ff ff ff ff 7f 78 78 78 78 78 78 78 78
0030	78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
0040	78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
0050	78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
0060	78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
0070	78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
0080	78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
0090	78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
00a0	78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
00b0	78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
00c0	78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
00d0	78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
00e0	78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
00f0	78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
0100	78 78 78 78 78 78 78 78 78 78 78 78 78 78 78

Fig. 6-3. Captura de datos mediante USRP, GNU Radio y Wireshark.

Los experimentos se monitorearon mediante Iperf, variando las velocidades de transmisión en las interfaces creadas para la conexión entre computadora y USRP. Los resultados en consola se muestran Fig. 6-5 y Fig. 6-6.

```

-----
Client connecting to 192.168.10.15, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[ 3] local 192.168.10.15 port 45318 connected with 192.168.10.15 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec
[ 3] Sent 893 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec    0.020 ms    0/ 893 (0%)
gibran@gibranHP:~$ iperf -c 192.168.10.15 -u
-----
Client connecting to 192.168.10.15, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[ 3] local 192.168.10.15 port 39669 connected with 192.168.10.15 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec
[ 3] Sent 893 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec    0.061 ms    0/ 893 (0%)
gibran@gibranHP:~$

```

Fig. 6-4. Monitoreo de características de transmisión en Iperf.

```

-----
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[ 3] local 192.168.10.15 port 5001 connected with 192.168.10.15 port 48530
[ ID] Interval      Transfer    Bandwidth      Jitter  Lost/Total Datagrams
[ 3] 0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec  0.017 ms    0/ 893 (0%)
[ 4] local 192.168.10.15 port 5001 connected with 192.168.10.15 port 45318
[ 4] 0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec  0.020 ms    0/ 893 (0%)
[ 3] local 192.168.10.15 port 5001 connected with 192.168.10.15 port 39669
[ 3] 0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec  0.062 ms    0/ 893 (0%)

```

Fig. 6-5. Monitoreo de características de transmisión en Iperf.

Capítulo 7

EXPERIMENTO DE VARIACIÓN EN MODULACIÓN

7.1. Procedimiento

Para el desarrollo del experimento se ocuparon las herramientas listadas anteriormente con la intención de hacer una comunicación donde la transmisión y la recepción funcionen con el mismo principio de operación y la comunicación sea exitosa, esta comunicación fue unidireccional y contaba con las siguientes características:

- Ganancia de Recepción y Transmisión de la antena 31.5 [dB]
- Ancho de banda de la aplicación en GNU Radio 5 [MHz]
- Offset del oscilador local de 11 [MHz]
- Longitud de paquete de 1500 [Bytes]
- Intervalo entre paquetes de 200 [ms]

En Fig. 7-1 se presenta la interfaz desplegada en la compilación de IEEE 802.11 en GNU Radio para recepción, dentro de ésta se observa cómo se pueden variar los parámetros del protocolo, además en Fig. 7-2 se encuentra la interfaz de transmisión.

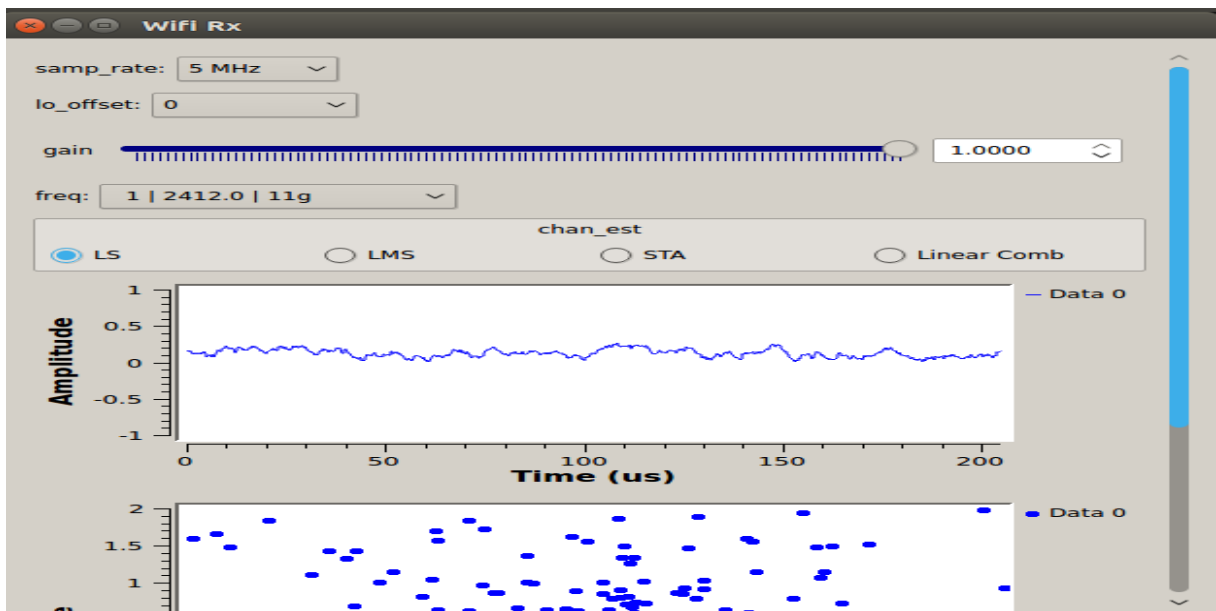


Fig. 7-1. Interfaz de Recepción.

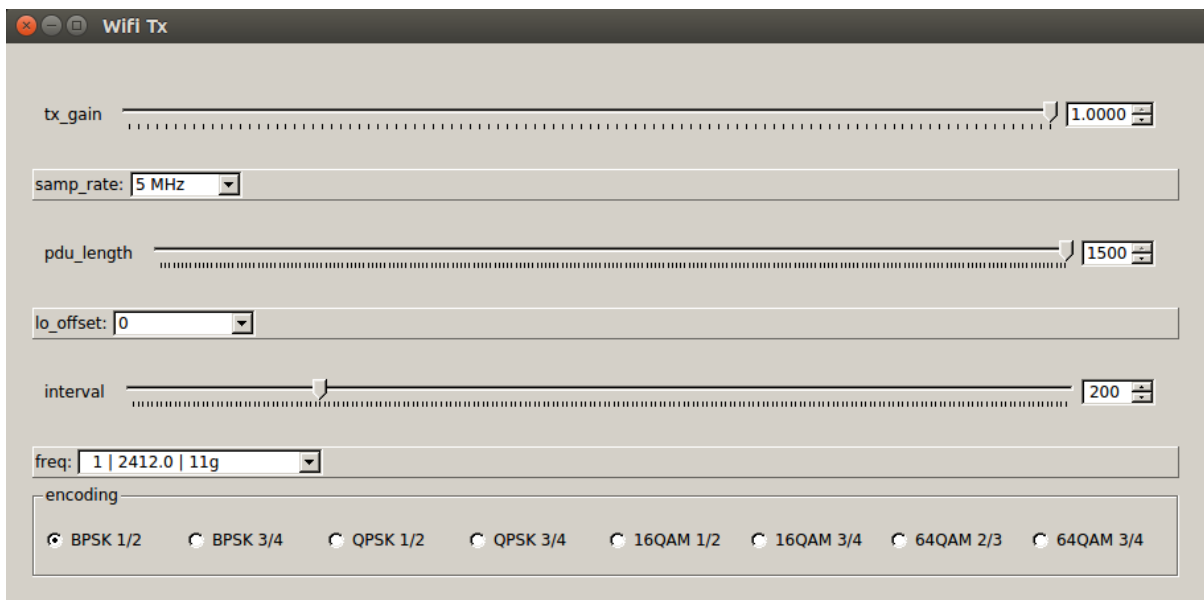


Fig. 7-2. Interfaz de Transmisión.

Para cada opción de modulación de la interfaz de transmisión mostrada en Fig. 7-2 se muestran las constelaciones formadas en Fig. 7-3, cada vez que la modulación se vuelve más compleja existe mayor Interferencia Intersimbólica, causando dispersión en la constelación, como se muestra en Fig. 7-5. Además, en Fig. 7-6 se muestra la comparación de la constelación entre un AP y un USRP con esquema de modulación 64QAM.

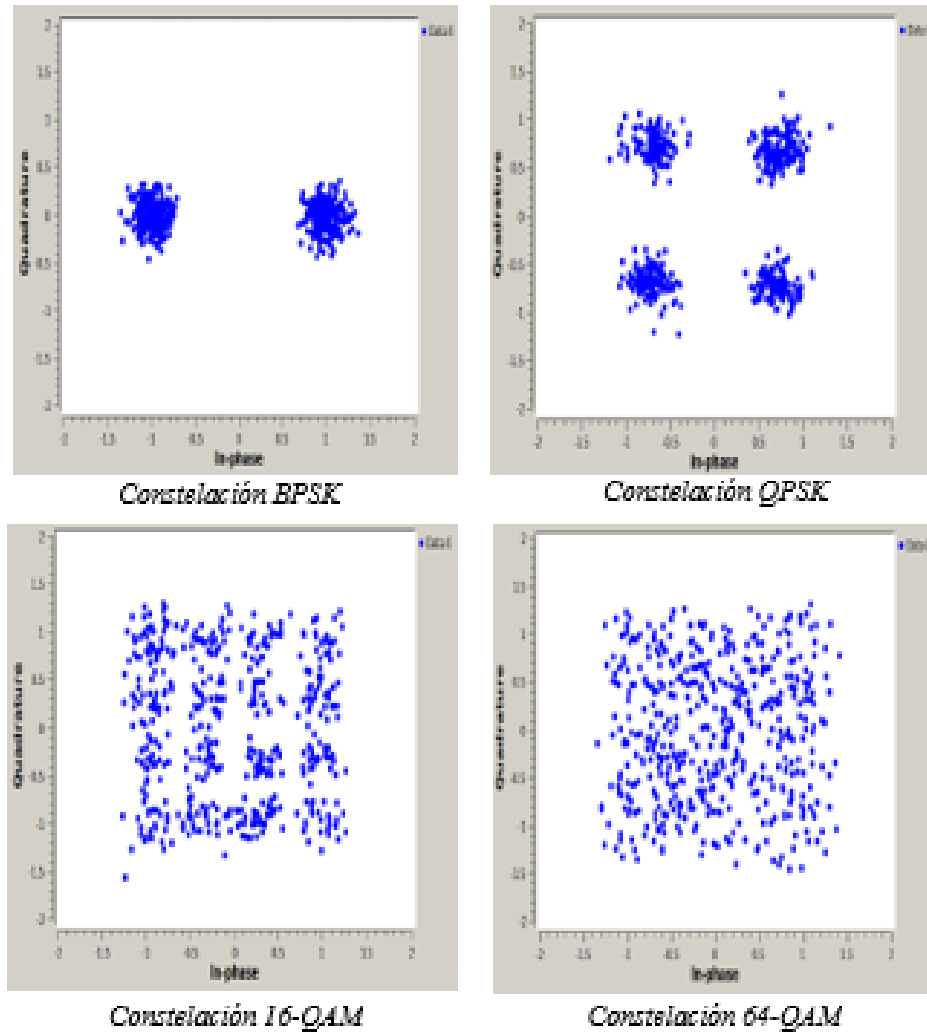


Fig. 7-3. Constelaciones GNU Radio.

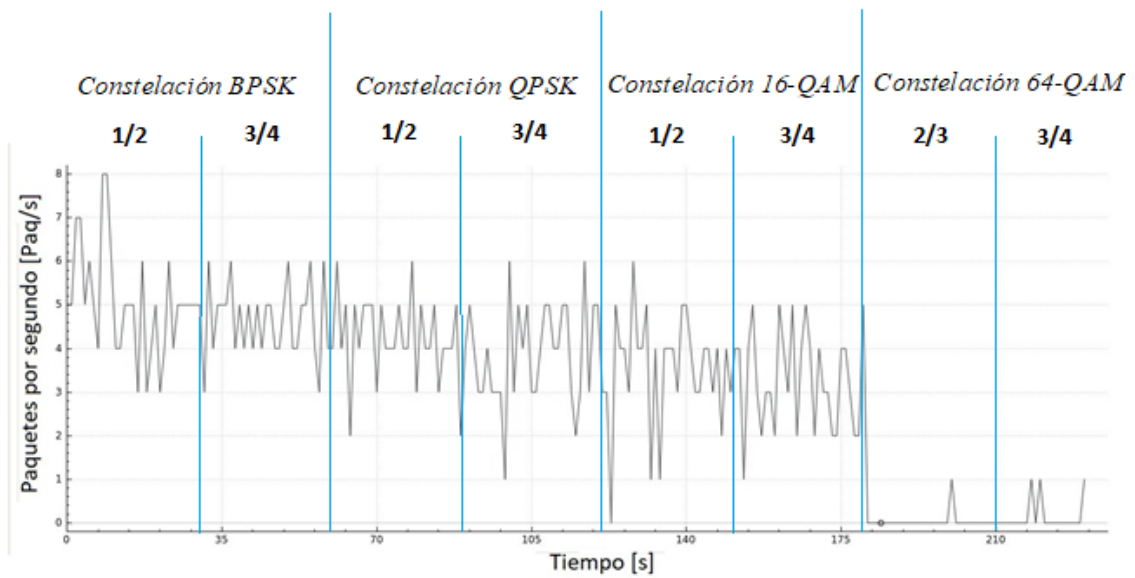


Fig. 7-4. Gráfica de paquetes recibidos variando la modulación.

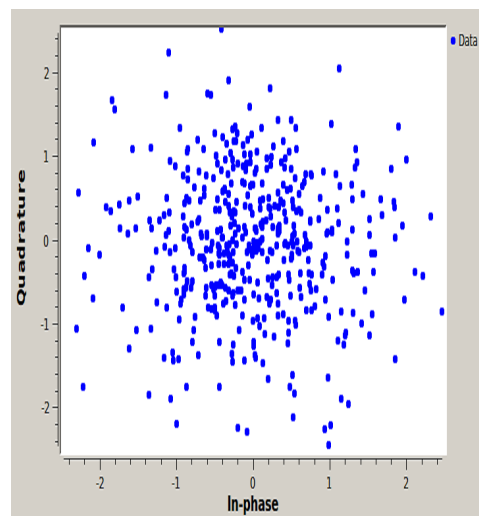


Fig. 7-5. Dispersión en constelación de recepción

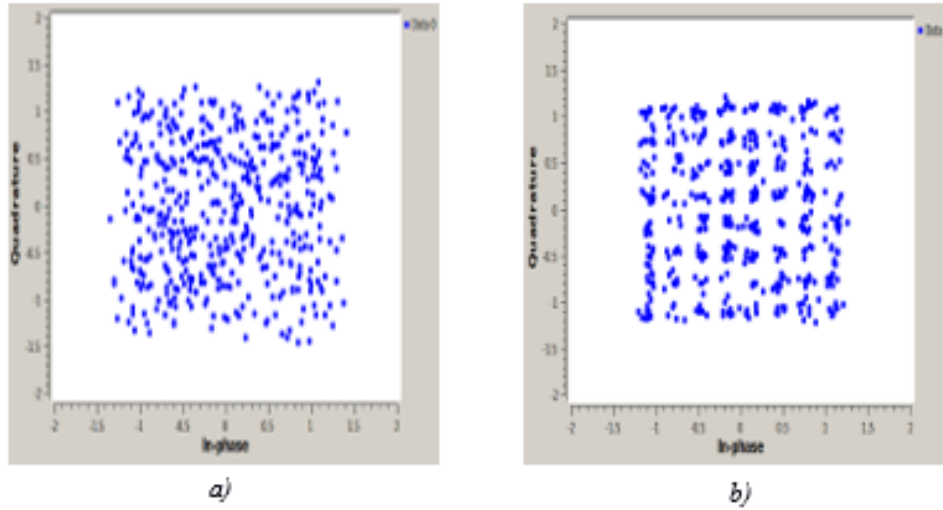


Fig. 7-6. Comparación entre AP y USRP con modulación 64QAM. a) Constelación por USRP, b) constelación por AP.

7.2. Resultados

Se varió el tipo de modulación durante cada 30 segundos, de esta manera, se apreciaría con mayor claridad la cantidad de paquetes recibidos con cada modulación. Cuando la modulación se vuelve más compleja hay menor cantidad de ceros entre muestras debido al incremento de pulsos en la transmisión y entre cada prefijo cíclico, lo que causa menor separación entre pulsos y a su vez la posibilidad de traslape entre cada pulso enviado. Cuando los datos codificados por los pulsos llegan al receptor hay Interferencia Intersimbólica entre estos [8]. En Fig. 7-7 se traza la gráfica del porcentaje exitoso de paquetes recibidos mediante la transferencia, el máximo se muestra en el 80 % y el promedio se encuentra en el 48.67 %.

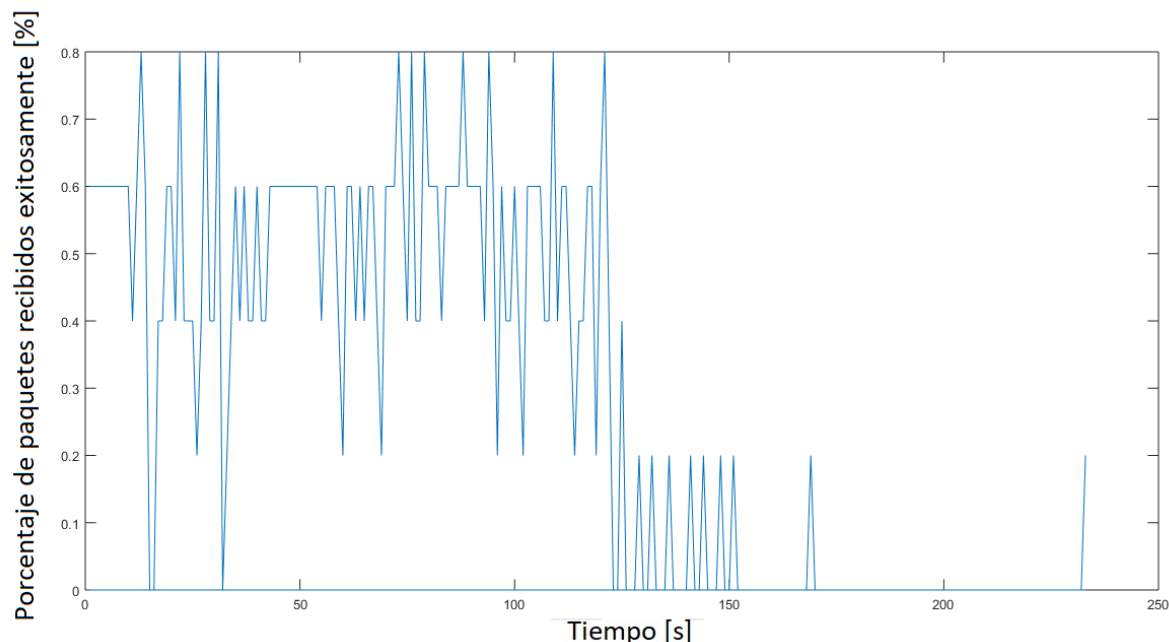


Fig. 7-7. Porcentaje de paquetes recibidos exitosamente con variación de modulación.

7.3. Análisis

En el primer minuto de la simulación se realizó una transmisión con constelación BPSK entre los dos radios USRP's. Los primeros 30 segundos fueron con una tasa de codificación Hamming $1/2$ ⁷ y los siguientes 30 segundos fueron con codificación $3/4$. En Fig. 7-7 se observa menor estabilidad en la gráfica de paquetes por segundo, es decir, no se muestra una variación pequeña en la recepción de paquetes en los primeros 30 segundos cuya codificación era de $1/2$, en lugar de ello se muestra una variación que, incluso, llega a cero en determinado segundo de la recepción. Con codificación $1/2$ la mitad de los datos son redundantes, por lo que se genera una saturación en el procesamiento más constante en los paquetes al transmitir y recibir que cuando se tiene una codificación mayor, causando inestabilidad en la gráfica de recepción de paquetes por segundo. En el segundo minuto de simulación se cambió a una transmisión QPSK, de los cuales, los primeros 30 segundos de ese minuto fueron dedicados a una codificación $1/2$ y los siguientes con codificación a $3/4$. La variación de recepción de paquetes en la modulación QPSK es menor

⁷El trabajo se desarrolló con 4 tipos de modulación BPSK, QPSK, 16QAM y 64QAM. Cada modulación contaba con dos opciones de codificación de tipo Hamming, estas eran $1/2$ y $3/4$.

Cuadro 7-1: Porcentaje promedio de paquetes recibidos en diferentes modulaciones.

MODULACIÓN	PORCENTAJE PROMEDIO DE PAQUETES RECIBIDOS
BPSK 1/2	48.67 %
BPSK 3/4	50.16 %
QPSK 1/2	54.67 %
QPSK 3/4	47.33 %
16QAM 1/2	9.33 %
16QAM 3/4	1.33 %
64QAM 2/3	0 %
64QAM 3/4	0.082 %

a la de BPSK con codificación 1/2, atribuyendo al hecho de que la distancia entre los radios fue idónea para trabajar con USRP's ⁸. Por otro lado, la recepción de paquetes a una codificación 3/4 con modulación BPSK tiene menor variación en la gráfica de paquetes recibidos por segundo en comparación con la modulación QPSK 1/2 y BPSK 1/2, esto es causado por el mismo motivo de la relación distancia-modulación. En el tercer minuto se modificó la modulación a 16QAM con las mismas dos codificaciones anteriores dividido, de la misma forma, en 30 segundos, el resultado fue el decremento de la recepción de paquetes. En la primera codificación hay más cantidad de paquetes recibidos que en la segunda codificación. Por último, en el cuarto minuto de simulación se cambió la modulación a 64QAM, cuyo porcentaje de recepción de paquetes fue de aproximadamente 0.04 %, con tan sólo un paquete recibido durante la codificación a 3/4. A continuación en la Tabla 7-1 se muestra el porcentaje de paquetes recibidos exitosamente para cada modulación y cada codificación.

Al ser una transmisión directa solamente entre dos nodos (USRP's) se genera el máximo envío de paquetes utilizando cualquier técnica de acceso al medio, sin embargo, los resultados no reflejan un gran porcentaje de paquetes recibidos exitosamente. Esto hace suponer que el envío de paquetes es directo desde el CPU al USRP sin tener tiempos de acceso al medio que verifiquen los tiempos de sensado y horarios de transmisión. Un problema común en simulaciones con GNU Radio y dispositivos SDR, que explica el poco porcentaje de éxito en la transmisión directa, es la tasa de transmisión con la que se acopla a la computadora o al mismo protocolo

⁸Existe relación proporcional entre cantidad de paquetes recibidos exitosamente y distancia, ésta tiene mayor radio de alcance mientras la modulación es menos compleja.

que se desea implementar. En el presente trabajo, el protocolo IEEE 802.11 y la verificación de horarios de transmisión no son completamente compatibles ya que se trabaja con un USRP que no tiene FPGA y no se puede acceder directamente al manejo de procesamiento en velocidades que requiere el estandar IEEE 802.11. Teniendo dispositivos con un FPGA incluido y software que soporte el protocolo 802.11 como en [12], es posible realizar una transmisión con un mayor porcentaje de paquetes exitosamente recibidos. Cabe mencionar que la construcción de IEEE 802.11 se ha realizado en USRP's generación x300 mediante una aplicación utilizada con GNU Radio llamada RFNoC en [13]. En Fig. 7-8 se muestra la comparación de porcentajes de paquetes exitosamente recibidos, para cada modulación con diferente codificación.

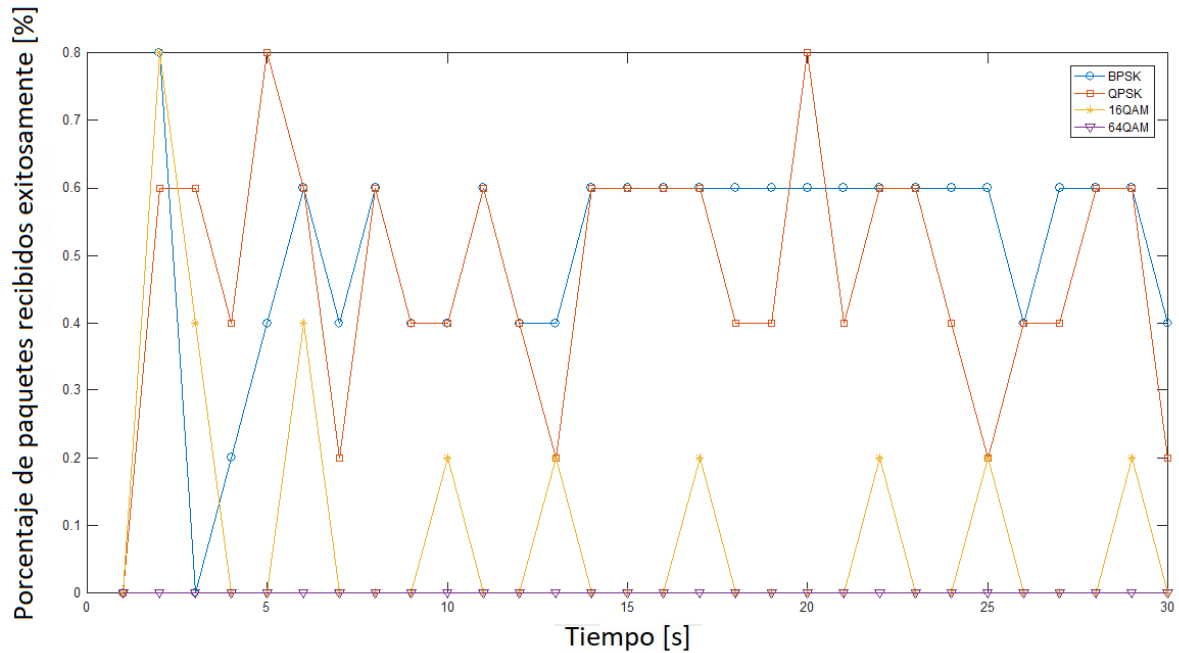


Fig. 7-8. 20 Porcentaje de paquetes recibidos exitosamente para todas las modulaciones.

Capítulo 8

EXPERIMENTO DE VARIACIÓN EN EL ANCHO DE BANDA

8.1. Procedimiento

Empleando la misma topología de Fig. 6-1, se modificó el ancho de banda de transmisión en las tres posibles opciones de transferencia de datos (5, 10 y 20 Megabits por segundo). Sólo en un ancho de banda de 5 [MHz] se presentó la comunicación entre los USRP's sin dispersión. También se realizó la simulación para un ancho de banda de 2 [MHz], sin embargo, no está estandarizado. En Fig. 8-1 se puede ver la gráfica de 5 [MHz]. Aunque la capa física en [7] haya sido implementada en GNU Radio como OFDM cuyas características, en un ancho de banda de 20 [MHz], son tener 64 portadoras, 52 portadoras útiles donde 4 de estas son piloto y su función principal de estas últimas cuatro es evitar el desfase con rutas de transmisión y controlar el ICI (Interferencia Cosimbólica), no puede emplearse una transmisión con ancho de banda de 20 [MHz].

8.2. Resultados

En Fig. 8-1 se muestra el porcentaje de paquetes efectivos respecto al ancho de banda de 5 [MHz]

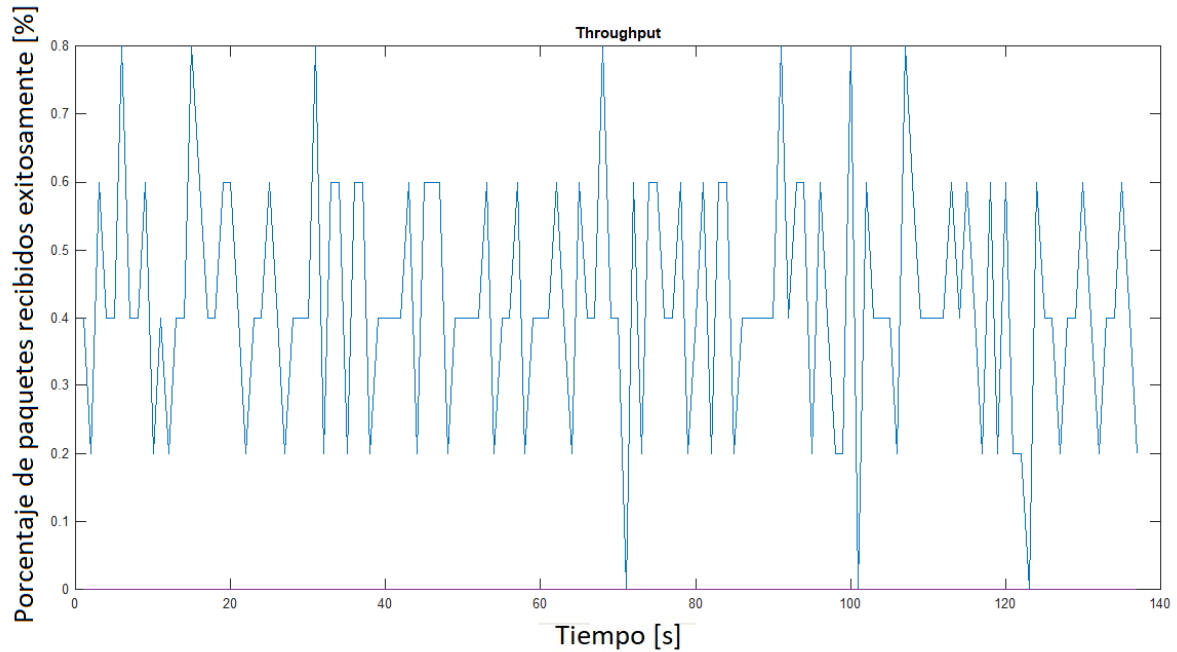


Fig. 8-1. .
]Porcentaje de paquetes recibidos exitosamente con ancho de banda de 5 [MHz].

8.3. Análisis

Al reducir el ancho de banda de transmisión se observa mayor cantidad de paquetes recibidos, dado que el espectro de frecuencia reduce, la velocidad de transmisión aumenta. La consecuencia de ello fue que al generar las pruebas con ancho de banda de 20 [MHz] o 10 [MHz] se obtuvieron en la captura por Wireshark paquetes malformados tal como se muestra en Fig. 8-2 debido a que la aplicación de GNU Radio y el USRP 2932 no soportaban la velocidad de transmisión.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000			WLAN	0	[Malformed Packet]
2	0.000000			WLAN	0	[Malformed Packet]
3	0.000000			WLAN	0	[Malformed Packet]
4	0.000000			WLAN	0	[Malformed Packet]

[Time delta from previous displayed frame: 0.000000000 seconds]					
[Time since reference or first frame: 0.000000000 seconds]					
Frame Number: 1					
Frame Length: 0 bytes (0 bits)					
Capture Length: 0 bytes (0 bits)					
[Frame is marked: False]					
[Frame is ignored: False]					
[Protocols in frame: radiotap]					
▼ [Malformed Packet: 802.11 Radiotap]					
▼ [Expert Info (Error/Malformed): Malformed Packet (Exception occurred)]					
[Malformed Packet (Exception occurred)]					
[Severity level: Error]					
[Group: Malformed]					

Fig. 8-2. Recepción de paquetes malformados.

La tasa de transmisión para los símbolos OFDM se obtienen con la ecuación mostrada en (8-1), rescatada de [14].

$$DataRate = \frac{1}{T_s} N_{FFT} m c c \quad (8-1)$$

$$DataRate = \frac{1}{\frac{(20)(10^6)}{64}} (44)(1)(1/2) = 5,5 [Mb/s] \quad (8-2)$$

Donde T_s es el tiempo del símbolo OFDM, N_{FFT} es el número de portadoras de datos, m es el número de bits por símbolo y cc es la tasa de codificación. El cálculo de la tasa de transmisión teórica en comparación con la experimental es diferente, puede ser causada por la misma aplicación. Al momento de pasar los datos de bloque a bloque existe la posibilidad de saturación antes de mandar desde la capa MAC virtual al USRP. El análisis de pase de mensajes desde la aplicación GRC por el lado de transmisión hasta la misma aplicación de lado de recepción se vuelve compleja ya que se necesita sincronización entre los dispositivos y horarios en los eventos de los bloques que sean adecuados con la sincronización.

Capítulo 9

EXPERIMENTO DE VARIACIÓN EN LA DISTANCIA

9.1. Procedimiento

Para el caso del experimento midiendo la distancia se trató de ir variando la distancia en un metro entre los radios USRP's hasta el punto donde ya no pudiera haber una constelación visible y en la recepción mostrara dispersión como en Fig. 7-5. Para ello, se fijó el ancho de banda, la modulación y el intervalo de transmisión de paquetes donde existiera mayor cantidad de paquetes recibidos y que hubiera menor variación en la gráfica de paquetes por segundo dadas las pruebas de verificación (se buscó la gráfica desplegada en Wireshark con distribución uniforme, sin algún punto donde la gráfica difiera en paquetes por segundo transmitidos). A continuación, se muestran las características establecidas:

- Ganancia de Recepción y Transmisión de 31.5 dB
- Ancho de banda de la aplicación en GNU Radio 5 [MHz]
- Offset del oscilador local de 11 [MHz]
- Longitud de paquete de 1500 [Bytes]

- Intervalo entre paquetes de 300 [ms]

9.2. Resultados

Se realizó el experimento primero con una medición del alcance máximo de comunicación entre los USRP's de 6.7 [m], después, se inició el experimento con una distancia entre radios de 10 [cm], variando cada minuto un metro hasta llegar a la distancia de 6.7 [m]. En Fig. 9-1 se muestra el porcentaje exitoso de paquetes recibidos de la distancia con un promedio del 40 %.

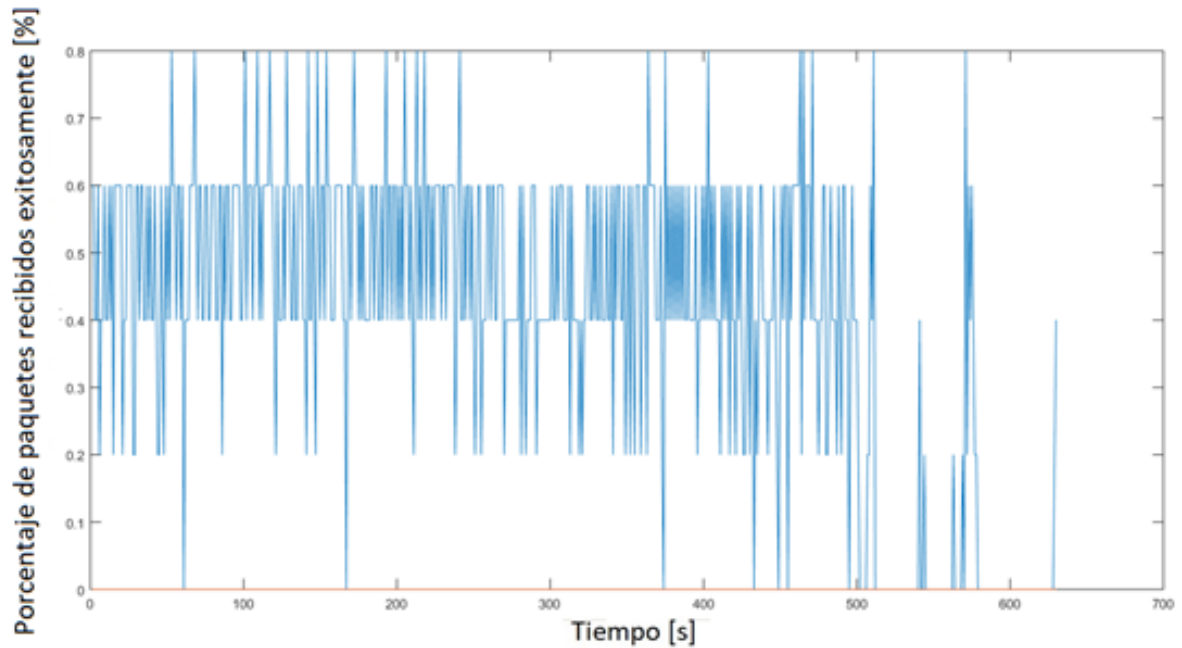


Fig. 9-1. Porcentaje de paquetes recibidos exitosamente con variación de distancia.

De igual manera, se generó una prueba a diferentes distancias, pero ahora con un AP como transmisor, se tomaron medidas de un metro de separación hasta notar en la interfaz de recepción de Fig. 7-5 dispersión en la constelación. Es difícil hacer una comparación entre los paquetes recibidos de un AP y los de un USRP ya que la transmisión de estos es diferente. Sin embargo, al variar la distancia, el rango de transmisión del AP resultó mayor que el de un USRP, con límite de 6.7 metros en la transmisión de un USRP y con más de 10 metros en la transmisión de un AP.

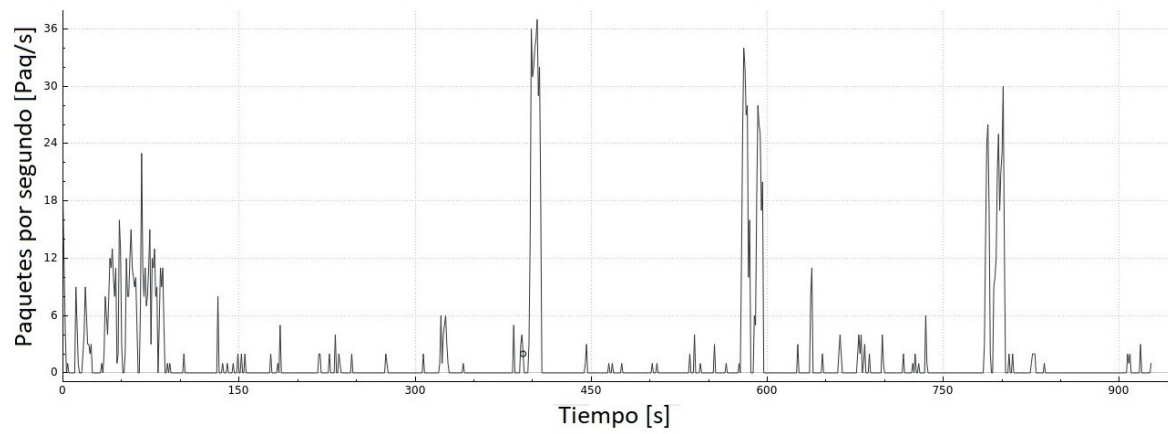


Fig. 9-2. Gráfica de cantidad de paquetes recibidos desde un AP con variación de distancia.

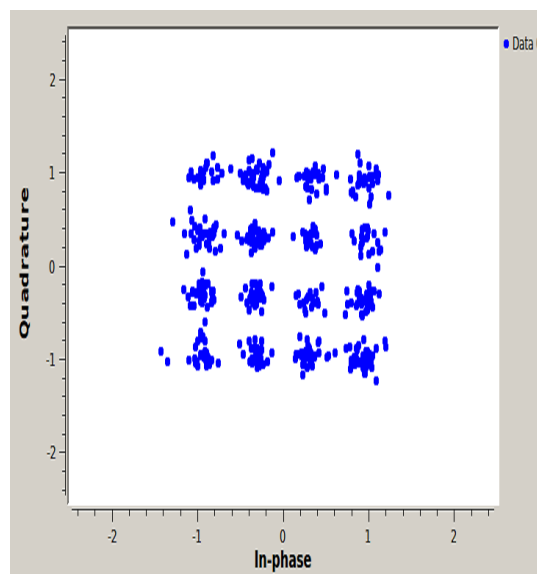


Fig. 9-3.]
Constelación de recepción de AP con ancho de banda de 20 [MHz]

9.3. Análisis

La constelación en la recepción en los primeros metros entre la transmisión del AP al USRP es mostrada en Fig. 9-3. El comportamiento de los paquetes recibidos se muestra en Fig. 9-2. Para los primeros minutos de simulación, cuando la separación entre AP y USRP todavía era menor a siete metros, la constelación recibida era de un esquema más complejo (NQAM) como en Fig. 9-2. Cuando se generaba una transmisión de AP a USRP que no fuera en línea de vista, mayor a siete metros, la constelación recibida era de tipo PSK. Entonces, se deduce que el AP acopla su RSSI a la distancia requerida. Una desventaja en la aplicación de GNU Radio es el no tener, como tal, ese cambio dinámico, sin embargo, es posible programarlo como trabajo futuro. Para el análisis de potencia y de distancia, se utilizó la fórmula general de PathLoss recuperada de [8] y los resultados obtenidos en el experimento.

$$P_r = P_t K \left[\frac{d_0}{d} \right]^\gamma \quad (9-1)$$

$$PL = K[dBm] - 10\gamma(\log_{10})\left[\frac{d_0}{d}\right] \quad (9-2)$$

$$K[dBm] = 20 \log_{10} \left[\frac{\lambda}{4\pi d_0} \right] \quad (9-3)$$

En la ecuación (9-2) $PL = P_r - P_t$, d_0 es la distancia a la antena para campo lejano, γ es el exponente del PathLoss, éste típicamente se encuentra en los rangos de 2.7 y 3.5 en zonas urbanas y es adimensional, en México se han generado pruebas en [15] donde el valor oscila entre 2,13 y 2,5, K se encuentra en dBm y es una constante útil para la obtención de PathLoss, para obtener su valor se utiliza la ecuación (9-3). Existe cierta facilidad para la obtención de las características del USRP al ser desplegadas en consola, a continuación, en Tabla 6-1 y Tabla 6-2 se muestra las características del USRP 2932 de transmisión y recepción respectivamente. Dadas las características del USRP 2932 podemos rescatar los valores de potencia teóricos de acuerdo a [8], calcular la potencia de transmisión y la distancia esperada con el diseño del

hardware y comparar con la prueba experimental. A continuación, en la ecuación (9-4) y (9-5) se hace la sustitución de los valores utilizados en la ejecución del experimento $\lambda = \frac{c}{f}$, donde $f = 4,4GHz$ y $d_0 = 100[m]$, después, en la ecuación (9-6) y (9-7) se realiza el despeje del valor de la distancia para campo lejano y para campo intermedio respectivamente.

$$K = 20\log_{10} \frac{3 \times 10^8}{4400000000} = -85,31 \quad (9-4)$$

$$K = 20\log_{10} \frac{3 \times 10^8}{4400000000} = -65,31 \quad (9-5)$$

Para campo lejano

$$d = ((PL - K))/(10 * 2,5) = 76,314[m] \quad (9-6)$$

Para campo intermedio

$$d = ((PL - K))/(10 * 2,5) = 9,607[m] \quad (9-7)$$

El PathLoss con los datos obtenidos fue expresado en la ecuación (9-4) y se obtiene el valor del experimento en la ecuación (9-8).

$$PL = -85,31[dBm] - 10(2,5)\log_{10}(100/6,7) = -114,658[dBm] \quad (9-8)$$

$$Pt = Pr - PL = 0[dBm] - (-47[dBm]) = 47[dBm] \quad (9-9)$$

En comparación con la potencia de transmisión mostrada en las características del USRP 2932, descritas en Fig. 9-4, la potencia de transmisión es 27 dB mayor a la potencia del resultado en (9-9). Tomando una Potencia mínima de recepción y una máxima en transmisión.

Capítulo 10

EXPERIMENTO DE VARIACIÓN DE TASA DE TRANSMISIÓN

10.1. Procedimiento

Para este experimento se realizó el incremento en el intervalo entre los paquetes transmitidos. Se realizaron 570 segundos de simulación, de los cuales, los primeros 30 segundos se fijó un intervalo entre paquetes de 10 [ms], después de cada 30 segundos se incrementó el intervalo 10 [ms] más, después de incrementar el intervalo entre paquetes a 100 [ms], los siguientes incrementos fueron de 100 [ms] cada 30 segundos hasta llegar al límite de 1000 [ms] entre paquetes transmitidos. Sus características fueron las siguientes:

- Ganancia de Recepción y Transmisión de 31.5 dB
- Ancho de banda de la aplicación en GNU Radio 5 [MHz]
- Offset del oscilador local de 11 [MHz]
- Longitud de paquete de 1500 [Bytes]

10.2. Resultados

Los resultados obtenidos se tuvieron que dividir en dos simulaciones, dado que, cuando se transmitían paquetes con un intervalo entre paquetes menor al tiempo de propagación más el tiempo de transmisión del paquete, el simulador empezaba a descartar paquetes para su recepción, además de tener una gran cantidad de paquetes en cola. En Fig. 10-1 se muestra la captura de paquetes en GNU Radio con Wireshark desde el transmisor, se observa una gran cantidad de paquetes a transmitir, sin embargo, durante la transmisión aparece en consola la leyenda “descartando paquetes”, esto verifica que la velocidad de transmisión del USRP limita la velocidad de transmisión de protocolo, causando el deshecho de paquetes a transmitir y la saturación del propio hardware. Incluso, en la recepción, la cantidad de paquetes recibidos con la transmisión de Fig. 10-2 es similar a la cantidad de paquetes recibidos de una transmisión donde el tiempo en el intervalo de paquetes es mayor al tiempo de propagación más el tiempo del paquete.

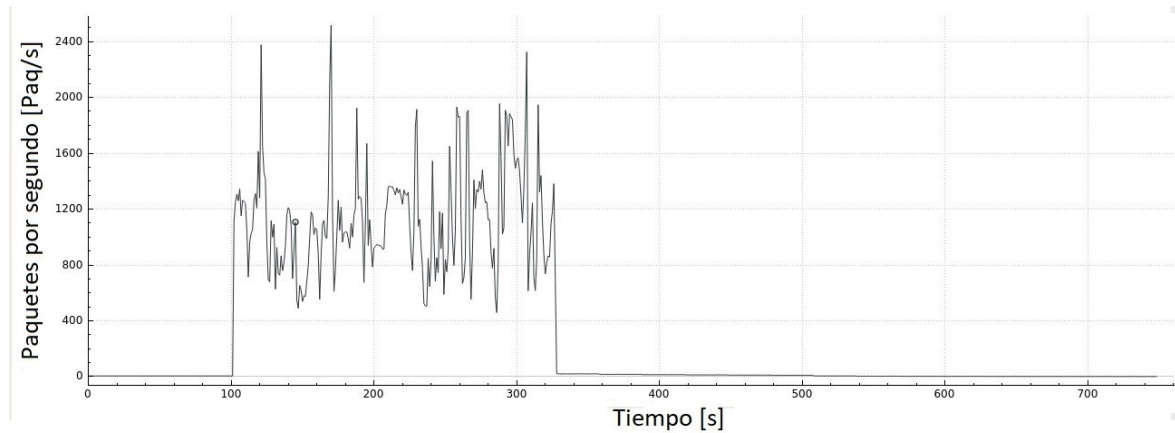


Fig. 10-1. Gráfica de cantidad de paquetes recibidos con variación de intervalo entre paquetes antes de 100 [ms].

En la gráfica de Fig. 10-2 se aprecia mejor la variación de paquetes al modificar el intervalo entre estos. Cada *escalón* que decrece en la gráfica de Fig. 10-2 es una variación del retardo entre paquetes.

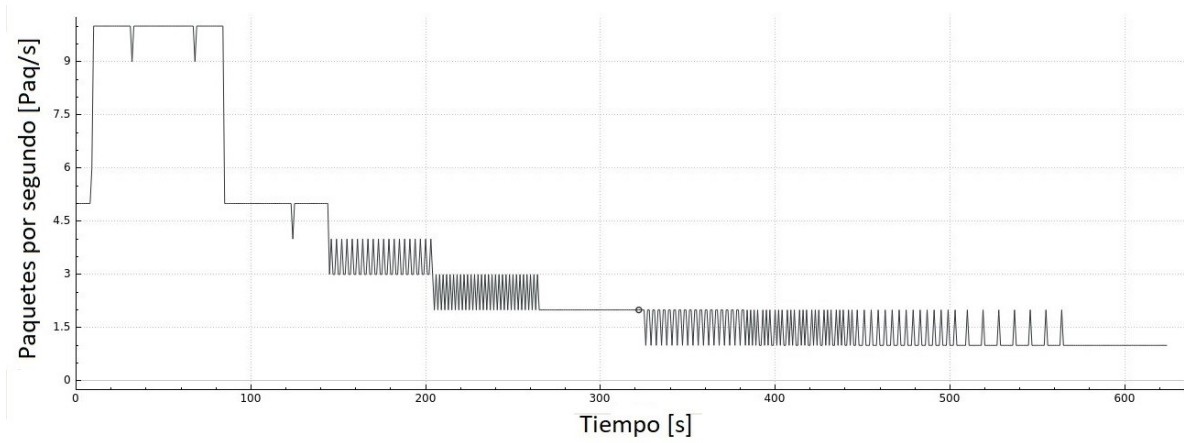


Fig. 10-2. Gráfica de cantidad de paquetes transmitidos con variación de intervalo entre paquetes después de 100 [ms]

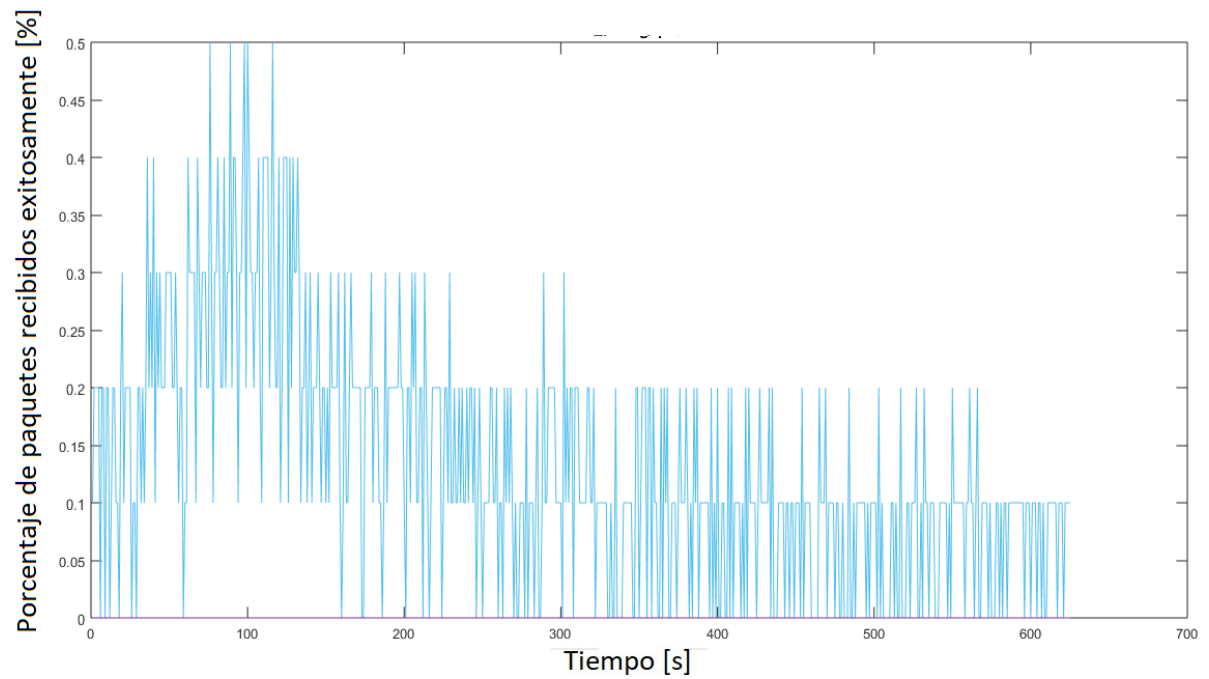


Fig. 10-3. Porcentaje de paquetes recibidos exitosamente con variación en Delay.

Cuadro 10-1: Comparación de throughput entre métricas.

EXPERIMENTO	MÁXIMO THROUGHPUT	MÍNIMO THROUGHPUT	THROUGHPUT PROMEDIO
Modulación	80 %	0 %	40.38 %
Ancho de Banda	80 %	0 %	9.54 %
Distancia	80 %	0 %	10.72 %
Retardo	80 %	0 %	17.98 %

10.3. Análisis

La gráfica en Fig. 10-2 muestra el desempeño de la transmisión a diferentes tiempos de envío de paquetes, cuando el comportamiento es más estable se puede inferir que se encuentra en su óptima transmisión, ya que no hay interferencia ni tiempo de espera en cola, sin embargo, cuando hay variación o pequeños "picos" en la gráfica, puede asumirse que se acumulan los paquetes a transmitir. El comportamiento de la gráfica en Fig. 10-3 muestra el porcentaje de paquetes exitosamente recibidos. Tomando en cuenta el máximo porcentaje de paquetes exitosamente recibidos durante la transmisión, la gráfica en Fig. 10-3 se asemeja a los resultados de una gráfica con acceso al medio de tipo Aloha Ranurado ⁹ con un máximo porcentaje aproximado de paquetes exitosamente recibidos durante la transmisión de 36.8 %. En la Tabla 10-1 se muestra la comparación entre porcentaje promedio de paquetes exitosamente recibidos.

⁹Técnica de Acceso al medio; divide el tiempo en ranuras, transmite el paquete en la ranura siguiente del tiempo en el que termina de generar el paquete. [16].

Capítulo 11

CONCLUSIONES

En la comparación entre la constelación recibida de la transmisión de USRP mostrada en Fig. 7-6a) y la transmisión de un AP en Fig. 7-6b), es notoria la superioridad en la transmisión de un AP en comparación con la transmisión de un USRP tanto en cobertura como en cantidad de paquetes transmitidos, esto se verifica al tener los puntos de la constelación mejor definidos o con menor dispersión, esta dispersión regularmente es causada por problemas de comunicación como la Interferencia Intersimbólica y cuando el retardo de las multitrayectorias exceden el prefijo cíclico que evita el traslape de señales [8]. Al ver la diferencia entre constelaciones en Fig. 7-6 y los resultados obtenidos al variar la distancia en la transmisión entre USRP's, se puede intuir que el problema no se encuentra en la recepción del USRP. Entonces, se concluye que los USRP's 2932 no tienen una cobertura aceptable en transmisión para escenarios de más de siete metros de distancia aun cuando se encuentran en línea de vista. De igual manera, se verifican los problemas del procesamiento del software de la compatibilidad entre dispositivos y de la compatibilidad con el propio estándar. Por otro lado, la tasa de transmisión de 5 [Mbps] de IEEE 802.11 implementada en GNU Radio, difiere al ser monitoreada con la herramienta Iperf, con resultado de tasa de transmisión de 1 Mbps, incluso, al intentar modificar la tasa de transmisión en la interfaz de red. Cuando se pretendió incrementar la capacidad de transmisión, la constelación capturada tuvo dispersión en la interfaz de recepción como se muestra en Fig. 7-6, así mismo, la propia aplicación GRC mostraba errores respecto a la capacidad del dispositivo.

Por tanto, la capacidad soportada por la computadora y por el USRP fue de 1 Mbps, en contraste a la capacidad de transmisión teórica que fue de 5.5 Mbps. Al igual que en la cobertura y la velocidad de transmisión, el diseño de Bloessl tiene modulaciones con mejor rendimiento. Al realizar el experimento con variación en las modulaciones, la modulación QPSK con codificación de $1/2$ obtuvo la mayor cantidad de paquetes exitosamente recibidos en promedio, sin embargo, la más estable fue la modulación BPSK con codificación $3/4$. El código 3 en la capítulo 6 refleja el entramado secuencial de datos programado en el bloque Message Strobe de Fig 6-2, cuya función del bloque es la de definir el tiempo de envío de información y la propia información. El experimento de variación en la tasa de transmisión dio como resultado un porcentaje de paquetes recibidos exitosamente cuyo máximo porcentaje de paquetes recibidos exitosamente se aproxima a la técnica de acceso al medio Aloha Ranurado. Al obtener estos resultados se puede sustentar que la técnica Aloha Ranurado que funciona de manera secuencial como en el código 3, puede ser la técnica que utiliza el protocolo IEEE 802.11 implementado en GNU Radio. Después del bloque Message Strobe se completa la capa de enlace de datos del modelo OSI con un siguiente bloque que le asigna una dirección MAC virtual al dispositivo USRP. La razón de comparar el porcentaje de paquetes exitosamente recibidos entre experimentos fue la de identificar la métrica que afecta mayormente el estándar IEEE 802.11 en GNU Radio al sufrir algún tipo de problema o variación. Los resultados en la comparación de la Tabla 10-1 muestra que las métricas más relevantes en el estándar desarrollado es el ancho de banda. La segunda métrica con mayor injerencia es la distancia entre radios y, por último, la tasa de transmisión, que dista bastante del porcentaje de paquetes recibidos correctamente con los resultados del experimento de modulación. Con los resultados obtenidos se sugiere que la aplicación IEEE 802.11 en GNU Radio, para USRP's 2932, debe mejorar la velocidad de transmisión y la sincronización tanto de computadora a USRP como de la aplicación GNU Radio a USRP. Si, además, se requiere implementar servicios de internet orientados a una arquitectura, se debe hacer una mejora en la potencia y directividad de las antenas. Este trabajo tiene como finalidad dar un esquema general del protocolo IEEE 802.11 implementado en SDR con la aplicación GNU Radio, por tanto, se trata de abarcar un análisis en las capas más relevantes del modelo OSI para la

conexión de dispositivos y concluir en qué escenarios es posible su utilización. Por el porcentaje de paquetes recibidos exitosamente durante la transmisión, además de las dificultades de la tasa de transmisión y la adaptación entre dispositivos, se deduce que el protocolo IEEE 802.11 en GNU Radio no es posible utilizarse en redes centralizadas, es decir, es preferible su uso en ambiente de tipo IoT con poca tasa de transmisión y que en casos particulares funja como la representación de un “nodo sink” cuya funcionalidad es la de administrar mensajes de los nodos de una red Ad-Hoc.

Capítulo 12

TRABAJO FUTURO

Es muy importante proveer internet a dispositivos inteligentes de IoT que es una tendencia cada vez más presente en el desarrollo de redes inalámbricas. Si bien el estándar desarrollado en GNU Radio no tiene las características de transmisión deseadas es aún utilizado en redes de sensores y demás dispositivos que no trabajen con una alta transmisión de datos. Es por ello por lo que un desarrollo importante sería en las técnicas de acceso al medio como CSMA y SMAC. Además del seguimiento de pruebas respecto a la distribución de los paquetes y la programación de nuevos algoritmos que se acoplen tanto al hardware del radio como al escenario que se desea implementar. En consecuencia, a la fácil variación en sus métricas, otra posible sugerencia a desarrollar es la predicción de escenarios saturados en canales adyacentes dado el comportamiento de sus métricas, con apoyo de aprendizaje automatizado como se realizó en [17].

Bibliografía

- [1] IEEE 802.11A, "*Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: high-speed physical layer in the 5 GHz band*", IEEE Std 802.11a, 1999
- [2] J. MITOLA, "*Software radios-survey, critical evaluation and future directions*", in Telesystems Conference, 1992. NTC-92., National, 1992, pp. 13/15-13/23.
- [3] TANGUY RISSET, "*Introduction to GNU Radio MAC-TC*", Citi Laboratory, INSA de Lyon, May 25, 2016.
- [4] GNU RADIO FOUNDATION, INC., 2019. Disponible en <https://www.gnuradio.org/>.
- [5] CHIA-HORNG, "*On the design of OFDM signal detection algorithms for hardware implementation*", In IEEE GLOBECOM 2003, pages 596–599, San Francisco, CA, December 2003. IEEE.
- [6] P. F, A. C., D. V., P. C., G. Z., T. Z., F. R, "*IEEE 802.11p Transmission Using GNURadio*", University of Salento, 73100 Lecce, Italy. 2012.
- [7] B. BLOESSL, M. SEGATA, C. SOMMER, AND F. DRESSLER, "AN IEEE 802.11A/G/P OFDM RECEIVER FOR GNU RADIO", in ACM SIGCOMM 2013, 2nd ACM SIGCOMM Workshop of Software Radio Implementation Forum (SRIF 2013). Hong Kong, China: ACM, Aug. 2013, pp. 9–16.
- [8] M. ERGEN, "*Mobile Broadband Including WiMAX and LTE*", Springer Science+Business Media, Spring Street, New York, NY 10013, USA, 2009.

- [9] K. FAZEL, F KAISER, "MULTI-CARRIER AND SPREAD SPECTRUM SYSTEMS; FROM OFDM AND MC-CDMA TO LTE AND WiMAX", A John Wiley and Sons, Ltd, segunda edición, Southern Gate, Chichester, West Sussex PO19 8SQ, United Kingdom, 2008.
- [10] J. HÖLLER, "FROM MACHINE-TO-MACHINE TO THE INTERNET OF THINGS: INTRODUCTION TO A NEW AGE OF INTELLIGENCE.", Academic Press, pp 352, 2014.
- [11] GNU RADIO. Polymorphics Types. *Disponible en* [[https](https://gnuradio.org/doc/doxygen/page_ofdm.html) :
//gnuradio.org/doc/doxygen/page_ofdm.html].
- [12] G. AGULLO, J. R., B. C. Y J. G., "An IEEE 802.11 MAC Software Defined Radio implementation for experimental wireless communications and networking research". Wireless Days. Uwicore, Ubiquitous Wireless Communications Research Laboratory. 2010
- [13] R. ZITOUNIZ, H. BOUAROUA AND B. SENOUCI. "Hardware-Software Codesign for Software DefinedRadio: IEEE 802.11p receiver case study". Conference RICESS, ECE Paris-LACSC Laboratory, VEDECOM institute, 2017. 37 Quai de Grenelle, 75015, Paris, France.
- [14] Y. ABRAHANTES. 2017. "Implementación del protocolo 802.11p basado en SDR para los sistemas ITS", [Tesis para obtener el título de Maestro en Ingeniería], Facultad de Ingeniería, Universidad Nacional Autónoma de México. Ciudad de México, México.
- [15] J. FLORES. 2009. "Estudio de Modelos de Propagación para la Implantación de una Red WiMAX en 3.3GHz en el Estado de Colima". [Tesis para obtener el título de Maestro en Ciencias] Centro de Investigación Científica y de Educación Superior de Ensenada. Ensenada, Baja California, México.
- [16] M. SCHWARTZ. "Telecommunication Networks: Protocols, Modeling and Analysis", Addison-Vesley Publishing Company, Department of Electrical Engineering and Center for telecommunications Research Columbia University, 1988.

- [17] S. KAJITA, H. YAMAGUCHI, T. HIGASHINO, “*Throughput and Delay Estimator for 2.4GHz WiFi APs: A Machine Learning-based Approach*”, 8th IFIP Wireless and Mobile Networking Conference, University of California, Los Angeles, 2015.
- [18] G. HERÍQUEZ. 2009. “*ANÁLISIS DEL DESEMPEÑO DE LA CAPA FÍSICA BASDA EN OFDM PARA LAS REDES INALÁMBRICAS DE BANDA ANCHA, BASADAS EN EL ESTÁNDAR 802.16-2004*”, [Tesis para obtener el título de Maestro en Ciencias e Ingeniería en Computación], Instituto de Investigaciones en Matemáticas Aplicadas y en Sistema, Universidad Nacional Autónoma de México. Ciudad de México, México.