Universidad de Cuenca

Facultad de Ingeniería

Carrera de
Electrónica y Telecomunicaciones

Implementación del estándar IEEE 802.11.p sobre el dispositivo Hack RF ONE en un ambiente controlado

Trabajo de titulación previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones.

Autores :

Diego Xavier Guillín Jiménez — C.I. 010662581-7
diegoxavier001@gmail.com

Luis Fernando Naula Rojas — C.I. 030239545-4
luchonaula@gmail.com

Director :

Ing. Darwin Fabián Astudillo Salinas, PhD — C.I. 010390703-6

Co-Director :

Ing. Alcides Fabián Araujo Pacheco, MSc — C.I. 010235850-4

Cuenca - Ecuador

12 de abril de 2021

Resumen

Las redes vehiculares consisten en la interconexión de nodos fijos y móviles y se enfocan en aplicaciones que ayudan al control de tráfico, seguridad y confort. El estándar que contiene las especificaciones técnicas es Institute of Electrical and Electronics Engineers (IEEE) 802.11p. Muchos estudios sobre este estándar se han realizado solo en simulaciones debido al costo de hardware necesario. En este trabajo se implementa el estándar IEEE 802.11p en GNU's Not Unix (GNU) Radio sobre el dispositivo Software Definide Radio (SDR) HackRF One, debido a su bajo costo. La implementación consiste de la capa física y de control de acceso al medio. Además, se implementa una aplicación que emite un mensaje en *broadcast* en escenarios estáticos y móviles. La infraestructura a utilizar es de una Road Side Unit (RSU) (estación fija) montada en un semáforo y la On Board Unit (OBU) (estación móvil) sobre un vehículo. En el desarrollo de esta tesis se realizaron pruebas de laboratorio para analizar y comprobar las especificaciones del estándar, además se realizaron pruebas en escenarios físicos, un escenario idealizado y un escenario real. En este último se realizan pruebas estáticas y en movimiento. Los datos a utilizar para el envío y recepción corresponden al estado de un semáforo y varían según su disposición. Finalmente, las pruebas son realizadas utilizando diferentes esquemas de modulación, variaciones en la potencia de transmisión a diferentes distancias y velocidades. Se define una configuración de parámetros para la transmisión de la señal que encaja en el rango de trabajo establecido para el dispositivo HackRF ONE. Dentro de las pruebas realizadas, se elige un tipo de modulación y la tasa de transmisión más robusta para estos escenarios. Por otro lado, de los datos obtenidos se analizan los paquetes recibidos, el *throughput* y el retardo obtenido en los diferentes escenarios estáticos y móviles.

Palabras clave : IEEE 802.11p. VANET. HackRF One. RSU. OBU. SDR. Semáforo.

Abstract

Vehicular Ad Hoc Network (VANET) consists of interconnecting static and mobile nodes and focus on applications that support control traffic, safety and comfort. IEEE 802.11p is the standard containing the technical specifications. Moreover, many research studies on this standard have only been applied in simulations due to high hardware costs. This thesis work implements the IEEE 802.11p standard on GNU Radio on the HackRF ONE SDR device, due to its lower cost. In this context corresponding to the standard, the Physical Layer (PHY) and Media Access Control (MAC) layer will be implemented, and an application that broadcasts a message in static and mobile scenarios is deployed. Moreover, the infrastructure to be used is from a RSU (static station) mounted on a traffic light and the OBU (mobile station) mounted on a vehicle.

Laboratory tests that are carried out in this thesis development, they are made to analyze and check the specifications of the standard, also tests in physical scenarios are carried out, an idealized scenario and a real scenario. In this last, static and mobile tests are carried out. The data that will be used to send and receive, correspond to the status of traffic light and will vary according to its arrangement.

Finally, tests are carried out using different modulation schemes, variation in transmission power at different distances, and speeds. Also, a parameter setting is defined for signal transmission that fits into a reliable working range for the HackRF ONE device. Within the tests performed, the type of modulation and the most robust transmission rate for these scenarios are chosen. On the other hand, the data that was obtained was used to analyze the packets received, the throughput, and the delay in the different static and mobile scenarios.

Keywords : IEEE 802.11p. VANET. HackRF ONE. RSU. OBU. SDR. Traffic light.

Índice general

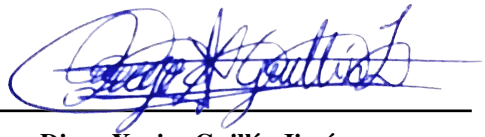
Índice de figuras

Índice de tablas

Cláusula de Propiedad Intelectual

Yo, Diego Xavier Guillín Jiménez, autor de la tesis "Implementación del estándar IEEE 802.11.p sobre el dispositivo Hack RF ONE en un ambiente controlado", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 12 de abril de 2021

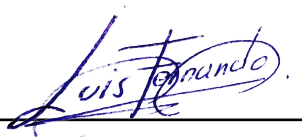


Diego Xavier Guillín Jiménez
010662581-7

Cláusula de Propiedad Intelectual

Yo, Luis Fernando Naula Rojas, autor de la tesis "Implementación del estándar IEEE 802.11.p sobre el dispositivo Hack RF ONE en un ambiente controlado", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 12 de abril de 2021



Luis Fernando Naula Rojas
030239545-4

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Yo, Diego Xavier Guillín Jiménez en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Implementación del estándar IEEE 802.11.p sobre el dispositivo Hack RF ONE en un ambiente controlado", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

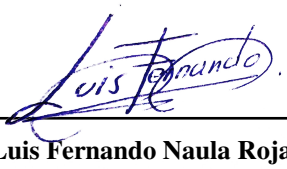
Cuenca, 12 de abril de 2021

Diego Xavier Guillín Jiménez
010662581-7

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Yo, Luis Fernando Naula Rojas en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Implementación del estándar IEEE 802.11.p sobre el dispositivo Hack RF ONE en un ambiente controlado", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 12 de abril de 2021



Luis Fernando Naula Rojas
030239545-4

Certifico

Que el presente proyecto de tesis: Implementación del estándar IEEE 802.11.p sobre el dispositivo Hack RF ONE en un ambiente controlado, fue dirigido y revisado por mi persona.

Firmado digitalmente por DARWIN FABIAN ASTUDILLO SALINAS
Fecha: 2021.04.12 07:01:19 -05'00'

Ing. Darwin Fabián Astudillo Salinas, PhD
Director

Certifico

Que el presente proyecto de tesis: Implementación del estándar IEEE 802.11.p sobre el dispositivo Hack RF ONE en un ambiente controlado, fue dirigido y revisado por mi persona.

Ing. Alcides Fabián Araujo Pacheco, MSc
Co-director

Dedicatoria

Este trabajo de titulación se lo dedico a mis padres, Claudio y Yolanda, quienes con su amor y ejemplo de trabajo y perseverancia me ayudaron a superar toda circunstancia y a cumplir esta meta. A mi hermana Cinthya que con su carisma y aprecio me inspira a ser su ejemplo de vida. A Daniela que con su cariño me ayuda a ser una mejor persona y se ha convertido en un pilar fundamental en mi vida. A todos mis familiares y amigos que han estado presentes a lo largo de mi vida estudiantil.

Diego Xavier Guillín Jimnénez

Dedicatoria

Este trabajo va dedicado para mi madre que con su amor, paciencia y sacrificio me ha ayudado a cumplir esta meta en mi vida, este logro se lo debo a ella. A mi pequeña y tan querida familia, gracias por sus consejos y apoyo incondicional en todos estos años de estudio. A todos mis amigos y compañeros con quienes he compartido bonitas anécdotas que serán difícil de olvidar.

Luis Fernando Naula Rojas

Agradecimientos

A los Ingenieros Fabían Astudillo, Alcides Araujo, director y co-director de este proyecto, por su apoyo y orientación en el desarrollo de este proyecto.

A todos los docentes de la Universidad de Cuenca, quienes nos han brindado su conocimiento y nos han motivado a lo largo de la carrera.

A nuestros compañeros y amigos formados en las aulas, por su aprecio y constancia en cada ciclo transcurrido.

LOS AUTORES

Abreviaciones y Acrónimos

AC Access Class. 36, 38
ADC Analog to Digital Converter. 39, 43, 118
AIFS Arbitration inter-frame spacing. 36, 37
AIFSN Arbitration Inter-Frame Space Number. 37
ARM Advanced RISC Machine. 117
AU Application Unit. 24

BPSK Binary Phase Shift Keying. 34–36, 57, 58, 62, 66, 69, 76, 77, 79, 82, 85, 86, 93, 94, 97, 99–101, 103
BSSID Basic Service Set Identifier. 39
BW Bandwith. 36, 37

CCH Control Channel. 23
CPLD Complex Programmable Logic Device. 76, 77, 93, 117
CSMA/CA Carrier Sense Multiple Access with Collision Avoidance. 36, 38
CSMA/CD Carrier Sense Multiple Access with Collision Detection. 38
CW Contention window. 37

DA Destiny Address. 39
DAC Digital to Analog Converter. 39, 118
DFT Discret Fourier Transformate. 33
DS Distribution Service. 38, 39
DSRC Dedicated Short Range Communication. 23, 29, 47

EDCA Enhanced Distributed Channel Access. 36
ETSI European Telecommunications Standards Institute. 23

FCC Federal Communication Commision. 23
FCS Frame Check Sequence. 38
FFT Fast Fourier Transform. 59

GNU GNU's Not Unix. 1, 2, 25, 27, 41, 43, 44, 46, 47, 50, 52–54, 71, 72, 74, 106, 113–115
GRC GNU Radio Companion. 116
GUI Graphical User Interface. 52, 115, 116

ICSP In Chip Serial Programmer. 118
IDE Integrated Development Environment. 40, 118
IEEE Institute of Electrical and Electronics Engineers. 1, 2, 23–25, 27–31, 34, 36, 38, 42, 43, 45–47, 49, 50, 52, 56, 61, 68, 71, 74, 75, 81, 102, 103, 105, 107

IFDT Inverse Fourier Discret Transformate. 32
ITS Intelligent Transportation System. 23, 25
ITU International Telecommunication Union. 25

LTS Long Training Sequence. 34

MAC Media Access Control. 2, 23, 27, 29, 34, 36–38, 43, 46, 52, 54–57, 62, 69, 70, 97, 102, 108
MSDU MAC Service Data Unit. 55

OBU On Board Unit. 1, 2, 24–29, 39, 42, 43, 45–50, 52, 76, 82, 83, 85, 86, 88–90, 92, 96, 98, 100–105, 108, 111
OFDM Orthogonal Frequency Division Multiplexing. 30–33, 43, 54, 59, 60, 62, 66, 67, 70, 74, 76–80, 82, 84, 85, 87, 90, 93, 98, 101–103
OSI Open System Interconnection. 23, 30

PHY Physical Layer. 2, 23, 27, 29, 33, 43, 46, 52, 54–57, 69, 70, 97, 102, 108
PLCP Physical Layer Convergence Procedure. 33, 34, 62
PLL Phase-Locked Loop. 39
PLME Physical Layer Managment Entity. 33
PMD Physical Medium Depend. 33, 34
PPDU PLCP Protocol Data Units. 34
PSDU Physical Service Data Unit. 55
PSK Phase Shift Keying. 35
PWM Pulse-Width Modulation. 118, 119

QAM Quadrature Amplitude Modulation. 32, 35, 36, 57, 58, 69, 76, 102, 103
QPSK Quadrature Phase Shift Keying. 32, 35, 36, 57, 58, 69, 76, 85, 86, 93, 94, 97, 99

RA Reception Address. 39
RSU Road Side Unit. 1, 2, 24–29, 39, 43, 45–49, 52, 75, 79, 82, 83, 85, 86, 88–90, 92, 95, 96, 98, 100–103, 105, 108, 111
RTP Real Time Transport Protocol. 105
RTS/CTS Request to Send and Clear to Send. 70

SA Source Address. 39
SCH Service Channel. 23
SDR Software Definide Radio. 1, 2, 25, 27, 28, 39–41, 43–45, 50, 70, 105, 106, 116
SMA SubMiniature version A. 40, 41, 43
STS Short Training Sequence. 34

TA Transmission Address. 39

UP User priority. 36
USB Universal Serial Bus. 109, 118, 121

V2I Vehicle to Infrastructure. 25, 29, 42
V2V Vehicle to Vehicle. 25, 29, 42

VANET Vehicular Ad Hoc Network. 2, 24, 28, 29, 42, 44, 45, 52

VNA Vector Network Analyzer. 42, 43, 71, 72, 120, 121

WAVE Wireless Access in Vehicular Environments. 24, 29

WEP Wireless Equivalent Privacy. 39

CAPÍTULO 1

Introducción

Este capítulo presenta los antecedentes, la identificación del problema, justificación, objetivos y estructura del presente proyecto.

1.1. Antecedentes

La estandarización de las redes vehiculares engloba desde la capa física hasta la capa de aplicación en el modelo Open System Interconnection (OSI). En 1999 la Federal Communication Commision (FCC) que representa a los Estados Unidos asignó una banda del espectro de radio de 75 MHz de ancho de banda en la frecuencia de 5,9 GHz para los servicios Intelligent Transportation System (ITS) que alberga de forma exclusiva las tecnologías de radiocomunicaciones para las redes vehiculares, el nombre que adopta el espectro de esta banda es Dedicated Short Range Communication (DSRC). En el caso de Europa, fue el European Telecommunications Standards Institute (ETSI) el que hizo lo propio, aunque la configuración fue realizada de forma diferente. En el estándar de los Estados Unidos, el espectro se estructura en siete canales de 10 MHz, como se muestra en la Figura 1.1, cada uno (172, 174, 176, 178, 180, 182 y 184). Se diferencian dos tipos de canales: Service Channel (SCH) y Control Channel (CCH) [1] [2].

Tanto en Estados Unidos como en Europa el IEEE decidió expandir la familia de estándares dedicados a las comunicaciones inalámbricas (IEEE 802.11) añadiendo el estándar IEEE 802.11p para dar soporte a las redes vehículares a nivel de las capas PHY y MAC usando las bandas que contemplaban las comunicaciones en la banda DSRC [1] [3] [4]. Este estándar cuenta con características particulares de transmisiones para tráfico vehicular, además, implementa una serie de mecanismos que lo hacen más idóneo que el 802.11 original [4].

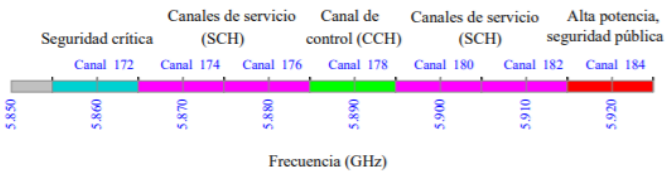


Figura 1.1: Estructura del espectro para 802.11p [4]

Cuando se habla de redes vehiculares es necesario mencionar a Wireless Access in Vehicular Environments (WAVE). Esta es la estandarización de un grupo de protocolos de la IEEE para el acceso inalámbrico en entornos vehiculares. WAVE describe la arquitectura de protocolos que administran las diferentes capas de nivel físico, acceso al medio, enlace y de red para las comunicaciones en redes ad hoc vehículares (VANETs) [1] [3].

Las VANETs son una especialización de las redes móviles ad hoc en donde cada vehículo se define como un nodo de la red, estos vehículos están equipados con una unidad de comunicación a bordo la que se denomina OBU y una unidad de aplicación denominada Application Unit (AU) [5] (Figura 1.2). La OBU tiene la función de intercambio de información con los demás vehículos o con los puntos de acceso ubicados a lo largo de las carreteras denominadas RSU. Una AU es la encargada de mostrar información para los usuarios, por ejemplo por medio de una pantalla o parlantes que se encuentren instalados en una OBU [5]. Al crear una red VANET los elementos que los conforman crean entre sí diferentes dominios o conjunto de elementos lógicos y físicos entre los que se encuentran [5] [6]:

- Dominio en el vehículo: Formado por la OBU y las AUs del vehículo que pueden conectarse alámbrica o inalámbricamente.
- Dominio ad hoc: Hace referencia a la comunicación inalámbrica que se usa para enlazar los nodos entre sí o los nodos con las RSUs.
- Dominio de infraestructura: Formado por las redes de acceso y la infraestructura que debe soportar el acceso a Internet que pueden solicitar los nodos y/o las RSUs.

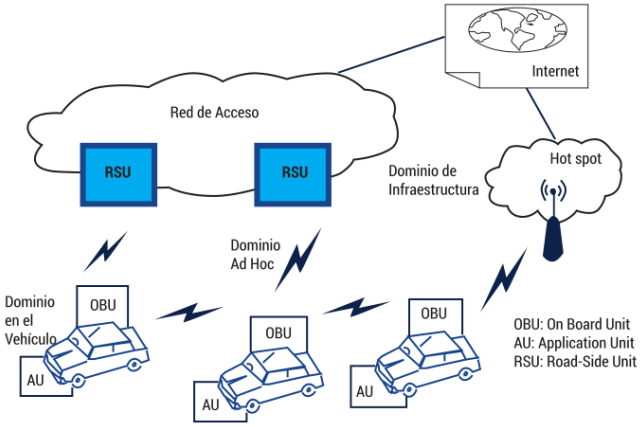


Figura 1.2: Componentes y diferentes dominios de una red VANET [3]

1.2. Definición del problema

A pesar de la mejora en la calidad de los vehículos y la infraestructura, los accidentes de tráfico y las muertes siguen siendo la principal preocupación. Los avances en las comunicaciones inalámbricas ofrecen varias vías interesantes para facilitar la implementación de nuevas funcionalidades vinculadas a este contexto. Son parte de un marco más grande conocido como ITS, y pueden integrarse en una infraestructura de telecomunicaciones existente o tener lugar directamente entre vehículos.

Las redes vehiculares conformadas por nodos fijos RSU o móviles OBU, interactúan en distintos escenarios: Vehicle to Vehicle (V2V), o Vehicle to Infrastructure (V2I) [7]. Además de las aplicaciones de seguridad, también permiten implementar aplicaciones de confort que pueden vincularse al contexto del vehículo (información de tráfico, atascos, espacios de estacionamiento) o no (acceso a Internet, juegos de red, etc.)[8].

1.3. Justificación

La tecnología para redes vehiculares se encuentra en desarrollo y el uso de *software* y *hardware* para la implementación de una aplicación enfocada a prevenir accidentes de tránsito o similares no son comunes debido a que conllevan costos elevados ya sea por el pago de una licencia o el uso de *hardware* hechos con un propósito específico. Es así que el uso de plataformas *open source* como lo es GNU Radio ofrecen una alternativa enfocada a la parte de *software* y complementando la parte de *hardware* con dispositivos *low cost* como lo es HackRF ONE; este equipo SDR tiene la capacidad de implementar distintas tecnologías dentro de un mismo dispositivo. Los SDRs permiten desarrollar un sistema de comunicación de manera eficiente y accesible.

1.4. Alcance

En este proyecto se propone la implementación del estándar IEEE 802.11p de la International Telecommunication Union (ITU) [9] sobre el dispositivo Hack RF ONE, el cual permite transmitir y recibir señales desde 1 GHz hasta 6GHz [9]. Este equipo viene integrado con una antena telescópica, diseñada a 50 Ohm para la transmisión y recepción [10]. El *software* se implementará sobre GNU Radio [11], obteniendo un código en el lenguaje de programación Python que se ejecutará sobre un sistema operativo Linux. Específicamente el estándar será probado en un ambiente controlado conformado por un transmisor (RSU) y un receptor que cumplirá el papel de la OBU. Durante el desarrollo de los experimentos, se variarán parámetros como ancho de banda, modulación y potencia de transmisión. La transmisión se realizará sobre el canal de control en la frecuencia de 5.89 GHz [12], utilizado para aplicaciones de seguridad. En la Figura 1.3 se presenta el esquema general del proyecto.

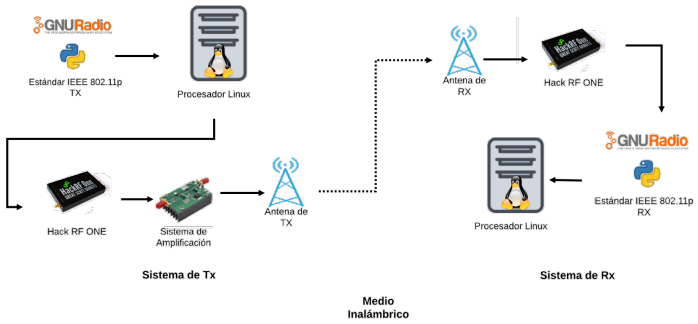


Figura 1.3: Esquema general del proyecto

El sistema de amplificación constará de un amplificador lineal con su propia alimentación. Se realizarán pruebas con la antena propia del Hack RF ONE y con antenas *microstrip* funcionando a la frecuencia que establece el estándar. El objetivo de la amplificación es contar con una mayor cobertura para la red ad hoc, en razón de que, en un principio el dispositivo Hack RF ONE no cuenta con una potencia de transmisión considerable para alcanzar grandes distancias de transmisión. El prototipo para el RSU será montado sobre una estructura fija a una altura similar a la de un semáforo de tránsito con el objetivo de realizar la transmisión de un mensaje de advertencia hacia el prototipo para el OBU. La comunicación entre los vehículos será automática con un mensaje de advertencia del tipo *broadcast* desde la RSU hacia las OBU conectados a la red ad hoc.

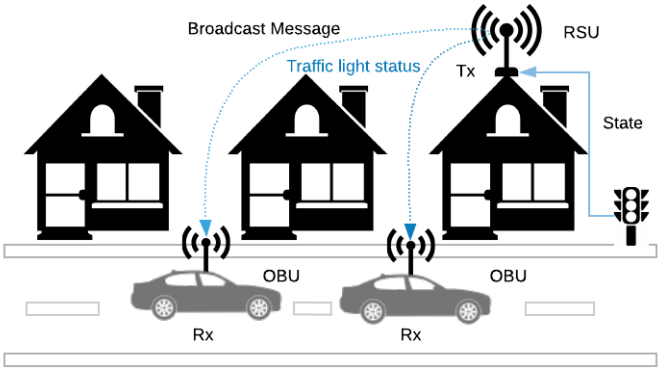


Figura 1.4: Infraestructura a implementar

Los prototipos serán montados, uno sobre un vehículo y el otro estará en una infraestructura. El sistema implementará una aplicación que enviará un mensaje. Las pruebas se realizarán como se indica en la Figura 1.4. Desde la RSU se enviará un mensaje en *broadcast* hacia el vehículo que se encuentra en la carreta. El mensaje indicará el estado en el que se encuentre un semáforo situado al final de la carretera. El mensaje emitido por la RSU se procesará en la OBU. La información recibida se dará a conocer por medio de una interfaz gráfica en el receptor. Las pruebas se realizarán con un vehículo en el cual estará integrado la OBU. La RSU enviará un mensaje en *broadcast* a las OBU, en las pruebas solo una OBU.

1.5. Objetivos

1.5.1. Objetivo general

Implementar el estándar IEEE 802.11p en GNU Radio sobre el dispositivo Hack RF ONE.

1.5.2. Objetivos específicos

El presente trabajo tiene los siguientes objetivos específicos:

- Implementar los bloques de capa PHY y MAC en GNU Radio.
- Implementar una aplicación que emita mensajes *broadcast* desde la RSU hacia la OBU.
- Montar el escenario físico para la comunicación entre la RSU y la OBU.
- Realizar pruebas de campo de transmisión y recepción.

1.6. Estructura de la tesis

El presente trabajo desarrolla en el Capítulo 2 el marco teórico sobre las redes vehiculares y el estándar IEEE 802.11p con su estructura conformada por la capa física, la capa de enlace de datos, el *software* y *hardware* utilizados. En el Capítulo 3 se detalla el medio de programación para dispositivos SDR, se presenta la arquitectura a implementar, sus componentes, se detalla las características y el funcionamiento de todo el sistema y se explica el montaje de los equipos. El Capítulo 4 explica el programa realizado en GNU Radio, el desarrollo de las capas que conforman el estándar IEEE 802.11p. El Capítulo 5 presentará el funcionamiento del dispositivo HackRF One. En el Capítulo 6 se encuentran los escenarios planteados, las pruebas realizadas y los resultados obtenidos. Finalmente se presenta las conclusiones, recomendaciones y trabajos futuros.

CAPÍTULO 2

Marco Teórico

El desarrollo de tecnologías y estándares para la comunicación inalámbrica permite a los usuarios varios mecanismos, servicios, aplicaciones y dispositivos evitando conexiones alámbricas. Entre los estándares más conocidos para redes inalámbricas están: el estándar IEEE 802.11 el cual especifica el nivel físico y de enlace de datos para una red de área local; el estándar IEEE 802.16 el cual está diseñado como una red de acceso, es más conocido como WiMax; y el estándar IEEE 802.15 el cual está especializado para redes de área personal, por ejemplo: Bluetooth y Zigbee.

La masificación de dispositivos móviles, el desarrollo computacional, y la capacidad en hardware, han permitido el crecimiento de las redes inalámbricas. Los escenarios en los que se implementan este tipo de redes pueden ser redes *mesh*, redes de sensores, redes aéreas, redes vehiculares, entre otras. Entre las aplicaciones en redes vehiculares que usan servicios *broadcast* se encuentran la gestión eficiente del tráfico, la prevención de accidentes, información del estado de las vías con el fin de prevenir percances, notificaciones sobre señales de tránsito, etc. Las redes vehiculares más conocidas como VANETs operan en varios entornos y bajo ciertos protocolos. Una VANET está compuesta por RSUs y OBUs las cuales intervienen en la comunicación. En la actualidad existen dispositivos especializados que funcionan como RSUs y OBUs, sin embargo, el costo de estos dispositivos es alto.

Los dispositivos SDR permiten realizar experimentación de cualquier tipo de protocolo. Estos dispositivos al ser equipos programables necesitan de software que cumplan esta función. La plataforma SDR y el *software* de programación usados en el trabajo de titulación son *open source* y *open hardware* respectivamente, y permite la implementación de protocolos. Para el desarrollo de una aplicación dentro de una VANET es necesario implementar los protocolos que involucran una VANET, y realizar varias pruebas de campo y laboratorio que validen los resultados. Para esto se cuenta con equipos de medición que facilitará el corroborar procedimientos.

Finalmente se aborda los trabajos relacionados al proyecto en curso, se presenta una recopilación de los principales trabajos en redes VANETs y las aplicaciones que se han desarrollado usando dispositivos SDR.

2.1. Redes vehiculares

Las redes vehiculares (VANETs) constituyen un tipo de red ad hoc inalámbrica. Una red ad hoc está formada por un conjunto de nodos móviles o fijos que pueden conformar una red temporal. Estas redes no usan una

infraestructura de red preexistente o un elemento central como puede ser un *access point* o *base station* [13].

Los nodos en una red ad hoc pueden tener movilidad, por lo tanto, su topología puede cambiar rápidamente y de forma impredecible; además, pueden estar constituidos por enlaces unidireccionales o bidireccionales.

Dentro de las redes ad hoc se puede establecer la comunicación entre dos nodos inalámbricos incluso cuando se encuentran fuera del rango de cobertura, gracias a que, entre ellos existe nodos intermedios que serán los encargados de reenviar los paquetes de datos desde la fuente hacia el destino. Este tipo de redes reciben el nombre de red inalámbrica multisalto [14]. Tomando en consideración las redes ad hoc pueden contar con conectividad a Internet mediante un elemento que haga las veces de *gateway*.

En las VANETs los nodos de la red son vehículos e infraestructura. Existen dos modos de comunicación en las VANETs, entre vehículos (V2V), y entre vehículos y estaciones a lo largo de la carretera (V2I).

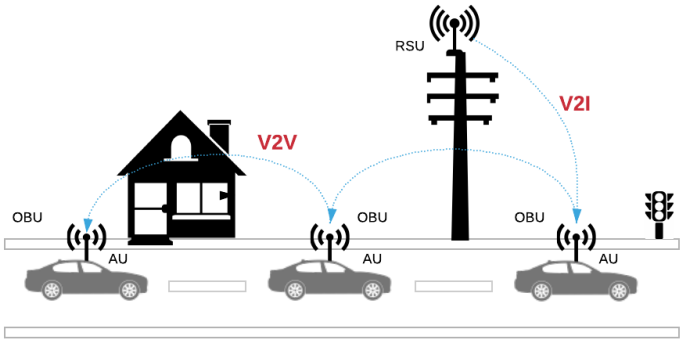


Figura 2.1: Modos de Comunicación en redes vehiculares

En la Figura 2.1 se analizan los modos de comunicación. En los vehículos se observa que existe un dispositivo denominado OBU, cuya función es intercambiar información entre vehículos que se encuentran en la carretera o con puntos de acceso estacionarios denominados RSU. Los dispositivos RSU son dispositivos de acceso ubicados a lo largo de las vías.

Las VANET poseen estándares que permiten el acceso y la comunicación V2V y V2I. Entre ellos está WAVE, el cual es un estándar basado en IEEE 802.11p designado para redes vehiculares. Mientras que DSRC establece el espectro y la banda que alberga a dichas redes.

2.1.1. WAVE

WAVE constituye la arquitectura de protocolos que administra la capa de nivel de red, enlace, acceso al medio (MAC) y física (PHY) para la comunicaciones en redes vehiculares. IEEE 802.11p es la propuesta realizada por la IEEE dentro de la arquitectura WAVE para la regulación de los mecanismos de acceso al medio en redes ad hoc vehiculares [4].

2.1.2. DSRC

La comisión federal de comunicaciones estadounidense estableció un espectro de 75 MHz en la banda de los 5.9 GHz para albergar de manera exclusiva las tecnologías que tendrían lugar en las redes vehiculares, es así que DSRC es el nombre que adopta el espectro en esta banda [15]. Este espectro se divide en 7 canales de 10 MHz cada uno (ver la Figura 2.3). La transmisión en DSRC se lleva a cabo a lo largo de cientos de metros, esta distancia es más corta que las que suelen soportar servicios celulares y WiMax [15]. En las capas PHY y MAC, DSRC utiliza el acceso inalámbrico IEEE 802.11p para entornos vehiculares (WAVE).

2.2. Estándar IEEE 802.11

El estándar IEEE 802.11 contiene una serie de especificaciones dedicadas a las redes inalámbricas. Las especificaciones contemplan la capa 1 (capa física) y capa 2 (capa de enlace de datos) del modelo OSI mostrado en la Figura 2.2 .

Figura 2.2: Modelo OSI

Dentro de IEEE 802.11 existen varias versiones las cuales se identifican por medio de una letra al final del nombre del estándar. Las versiones reflejan características diferentes como frecuencia de operación, alcance, ancho de banda, tasa de transmisión y máxima potencia de transmisión [16]. La Tabla 2.1 resume las características de cada versión [17]. Los estándares más recientes utilizan Orthogonal Frequency Division Multiplexing (OFDM), el cual mejora la eficiencia espectral, aumentando las prestaciones y permitiendo transmisiones de manera simultanea.

Tabla 2.1: Versiones del protocolo IEEE 802.11.

Estándar	802.11	802.11b	802.11a	802.11g	802.11n	802.11p
Frecuencia	2.4 GHz	2.4 GHz	5 GHz	2.4 GHz	2.4 y 5 GHz	5.9 GHz
Alcance	20 m	35 m	35 m	70 m	70 m	1000 m
Ancho de Banda	20 MHz	25 MHz	20 MHz	25 MHz	20 y 40 MHz	10 MHz
Tasa de Transmisión	2 Mbps	11 Mbps	54 Mbps	54 Mbps	600 Mbps	27 Mbps
Máxima Potencia	100 mW	100 mW	100 mW	100 mW	100 mW	760 mW

2.3. Estándar IEEE 802.11p

Una especificación de las redes ad hoc diseñada para las redes vehiculares es el estándar IEEE 802.11p [18]. Este protocolo opera en las capas 1 y 2 del modelo OSI que corresponden a la capa física y a la capa de acceso al medio respectivamente.

2.3.1. Capa física

En el estándar IEEE 802.11p utiliza OFDM como esquema de modulación, optimizando la tasa de transmisión por medio de sub-portadoras ortogonales en rangos de frecuencias delimitados. Se consideran rangos o bandas de frecuencias para aplicaciones específicas de redes vehiculares tanto para bandas en Estados Unidos como para Europa [12]. En la Figura 2.3 se observa la banda utilizada en IEEE 802.11p; esta va desde 5.850 GHz hasta 5.925 GHz. Cada canal existente tienen un ancho de banda de 10 MHz [19].

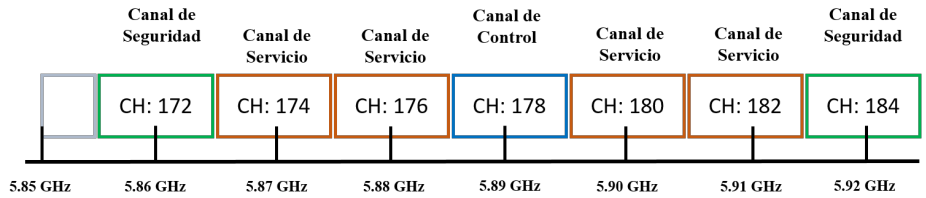


Figura 2.3: Espectro utilizado en el estándar IEEE 802.11p

Como se muestra en la Figura 2.3, el estándar 802.11p especifica 7 canales, utilizados para servicio, control y canales reservados para seguridad.

- **Servicio:** Utilizan los canales 174, 176, en las frecuencias: 5.87 GHz, 5.88 GHz y los canales 180 y 182 a 5.900 GHz y 5.910 GHz respectivamente. Estos canales pueden ser utilizados individualmente o combinarse formando un canal de 20 MHz.
- **Control:** Utiliza el canal 178 a 5.890 GHz, este se encarga de controlar la transmisión y el establecimiento del enlace.
- **Seguridad:** Utiliza el canal 172 a 5.860 GHz y el canal 184 a 5.920 GHz. Estos canales permiten soluciones de seguridad vial y se encargan de la gestión de tráfico en los demás canales.
- **Guarda:** Al comienzo de la banda, en la frecuencia 5.85 GHz, existen 5 MHz que se pueden utilizar como una banda de guarda [20].

En el canal de control los mensajes son de tipo *broadcast*, con alta periodicidad y baja latencia. Los canales 178 y 180 actúan en pro de la seguridad vial, mientras que los canales 176, 182, y 184 se enfocan en la eficiencia vial [21].

2.3.1.1. OFDM

La multiplexación ortogonal por división de frecuencias (OFDM) es un mecanismo utilizado en los estándares IEEE 802.11a/g/n, gracias a que permite modular un conjunto de datos en diferentes sub-portadoras ortogonales entre si evitando interferencias [22].

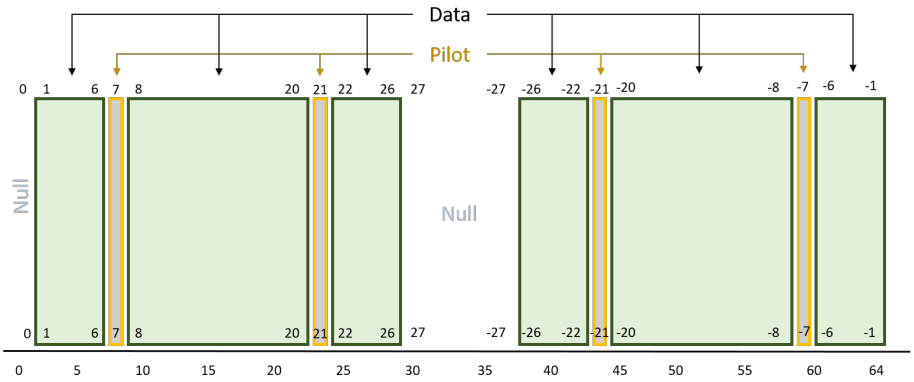


Figura 2.4: Sub-portadoras OFDM

En la Figura 2.4 se muestra la distribución de las sub-portadoras. Estas están divididas en sub-portadoras piloto, datos y nulas. De un total de 64 sub-portadoras se utilizan 52: 4 sub-portadoras piloto y 48 sub-portadoras de datos. Las sub-portadoras piloto ayudan a rastrear desplazamientos de frecuencia y ruidos de fase, estas son la 7,21,-21,-7. Las 48 sub-portadoras de datos van de 1 - 6, 8 - 20, 22 - 26, -26 - -22, -20 - -8, -6 - -1. Las sub-portadoras nulas o de relleno ocupan las posiciones 0 y desde la 27 hasta 37 [19].

En la Figura 2.5 se muestra el diagrama de bloques de un sistema OFDM. El proceso de trasmisión es el siguiente:

1. El bloque S/P toma de entrada una trama de bits de información (en serie)($\overrightarrow{b}$) y la trasforma en símbolos en paralelo acorde al tipo de modulación que se utilice: Quadrature Phase Shift Keying (QPSK) (2 columnas por 2 filas), 16-Quadrature Amplitude Modulation (QAM) (4 columnas por 4 filas), y 64-QAM (8 columnas por 8 filas).
2. Los bits en paralelo se envían hacia el bloque de *Mapping*. Este bloque modula los bits usando los sistemas QPSK, 16-QAM o 64-QAM. En la salida de este bloque se forman los símbolos.
3. A cada símbolo resultante se le aplica el bloque Inverse Fourier Discret Transformate (IFDT) para que la señal de las portadoras sea ortogonal en tiempo y frecuencia.
4. La señal que sale del bloque IFDT es convertida en serie con el bloque P/S.
5. El prefijo cíclico se añade a la señal OFDM durante un tiempo Δ, llamado intervalo de guarda. Debido a ello la duración del pulso enviado realmente es $T_s = T + \Delta$. El intervalo de guarda es una característica importante de OFDM que permite neutralizar ecos y fallos de sincronización temporal inferiores a δ segundos. Con la introducción del prefijo cíclico se reduce la eficiencia energética y la tasa binaria transmitida [23].
6. La señal resultante es enviada a través de un canal inalámbrico móvil donde señal OFDM se ve afectada por ruido Gaussiano y desvanecimiento multicamino del tipo Rayleigh o Ricean.

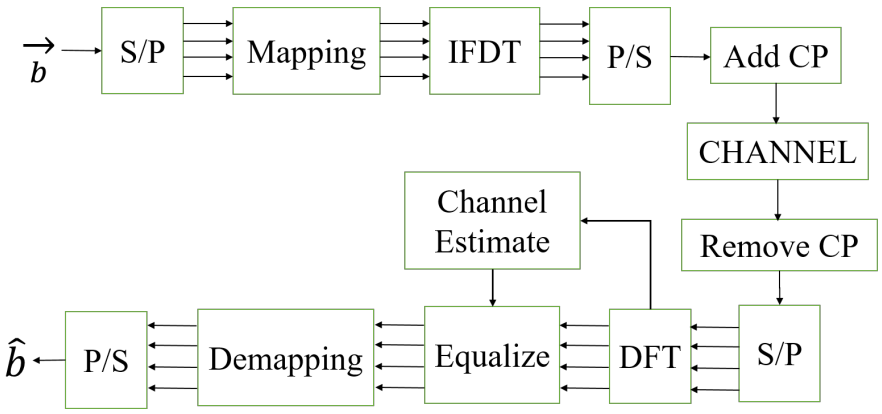


Figura 2.5: Modelo OFDM [23].

En la Figura 2.6 se muestra el espectro de una señal OFDM, la cual está formada por varias sub-portadoras, las señales no se intervienen en el dominio de la frecuencia gracias a la ortogonalidad presente, siendo estas linealmente independientes. OFDM se puede adaptar en canales de distintos anchos de banda a demás de ser muy robusta contra la interferencia inter-símbolo [24].

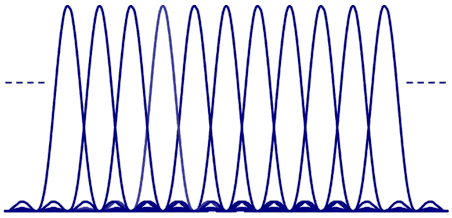


Figura 2.6: Espectro OFDM [25].

Al recibir la señal se aplica el inverso al proceso de trasmisión de la siguiente manera:

1. El bloque Remove CP elimina el prefijo cíclico de la señal que es adquirida del medio de transmisión.
2. La señal sin el prefijo cíclico es convertida en paralelo en el bloque S/P.
3. Se aplica la transformada de Fourier a la señal por medio del bloque Discret Fourier Transformate (DFT). Como la señal OFDM es un conjunto de sub-portadoras, la mayoría de estas se utilizan para los datos, mientras que otras permiten estimar el comportamiento del canal.
4. Las sub-portadoras que ayudan a la estimación del canal se llaman sub-portadoras piloto y permiten realizar la correcta estimación del canal y así poder equalizar la señal, eliminado los efectos del canal sobre nuestra señal recibida.
5. . Luego del proceso de estimación del canal y equalización la señal pasa al bloque Demapping, en donde la señal será demodulada dependiendo del esquema de modulación con la que se esta trabajando en el transmisor.
6. Los símbolos que se obtienen luego de la demodulación se convierten en una trama de bits por medio del bloque P/S y así finalmente obtenemos la información enviada desde el transmisor ($\hat{b}$) [22].

2.3.1.2. Estructura de la capa Física

La capa física o PHY está compuesta por tres subcapas: Physical Layer Managment Entity (PLME), Physical Layer Convergence Procedure (PLCP), Physical Medium Depend (PMD). La PLME es la entidad de

administración del medio físico, esta gestiona y administra las funciones de la capa física local con la de la capa MAC [12]. El PLCP es el procedimiento de convergencia de capa física, este adapta las capacidades para el servicio MAC, asigna las unidades de datos de forma correcta para la transmisión y recepción, entrega las tramas entrantes a la subcapa MAC y estructura los bits sumándole un encabezado, preámbulo y una cola, creando así el PLCP Protocol Data Units (PPDU), como se muestra en la Figura 2.7.

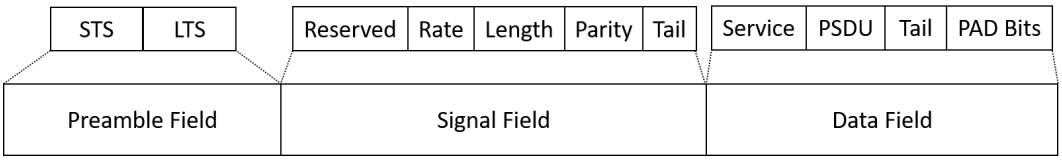


Figura 2.7: Estructura del Frame PPDU

La trama PPDU está compuesta por el campo **`Preample`**, **`Signal`** y **`Data`**, estos se detallan a continuación:

- **`Preamble:`** Está definido con 10 Short Training Sequence (STS) (símbolos cortos), y 2 Long Training Sequence (LTS) (símbolos largos) [12]. Los STS son los encargados de la detección de la señal, el control automático de ganancia, la selección de la diversidad, la compensación de frecuencia, y la sincronización [18]. Los LTS ayudan en la recepción de la trama, teniendo como prioridad estimar el canal.
- **`Signal:`** Permite la modulación Binary Phase Shift Keying (BPSK) a una tasa de 1/2. En el campo `Rate` está el esquema de modulación y la tasa de codificación para la información a trasmitir [18]. En el campo `Length` se indica el número de octetos. El siguiente campo indica la paridad. Los últimos bits son utilizados para sincronización [20].
- **`Data:`** Está compuesto por el campo `Service`, el cual permite la sincronización del proceso de aleatorización; el campo `PSDU`, el cual restablece a 0 el codificador convolucional [18]; y el campo `PAD`, el cual permite completar el número de bits del campo `Data`.

El PMD es el sistema dependiente del medio físico, este establece las características y métodos de envío y recepción de los datos. Está encargado de los parámetros de tipo físicos: tipo de señal, frecuencia, amplitud, modulación, entre otras [18].

2.3.2. Modulación

De acuerdo al ancho de banda de 10 MHz, en la Tabla 2.2 se presentan los posibles tipos de modulación del estándar IEEE 802.11p [21]. Este aplica las especificaciones del estándar IEEE 802.11 para el ancho de banda mencionado.

Tabla 2.2: Tipos de modulación para el estándar IEEE 802.11p [21].

Modulación	Tasa de codificación	Bits codificados por subportadora	Bits por símbolo OFDM	Bits de datos por símbolo OFDM	Tasa de transmisión
BPSK	1/2	1	48	24	3 Mb/s
BPSK	3/2	1	48	36	4.5 Mb/s
QPSK	1/2	2	96	48	6 Mb/s
QPSK	3/4	2	96	72	9 Mb/s
16-QAM	1/2	4	192	96	12 Mb/s
16-QAM	3/4	4	192	144	18 Mb/s
64-QAM	1/2	6	288	192	24 Mb/s
64-QAM	3/4	6	288	216	27 Mb/s

La modulación BPSK conocida como 2-Phase Shift Keying (PSK), consiste en el desplazamiento de fase para 2 símbolos. Es la más sencilla de emplear en razón de que utiliza 2 símbolos con 1 bit de información cada uno [26]. Estos símbolos tienen un valor de salto de fase de 0º para el 1 y 180º para el 0 (-1), como se muestra en el diagrama de constelación en la Figura 2.8(a). La modulación QPSK o *Quaternary* PSK contiene cuatro fases y puede codificar dos bits por cada símbolo [27]. En el diagrama de constelaciones (Figura 2.8(b)), QPSK representa cuatros puntos equidistantes al origen. La modulación M-QAM consiste en la modulación por desplazamiento de fase y amplitud, en donde M es el número de símbolos y debe ser número par. El número de bits se lo obtiene por medio de la relación: $M = 2^n$, en donde n es el número de bits por símbolo. Para el caso de 16-QAM se tiene 16 símbolos utilizando 4 bits. En 64-QAM se tiene 64 símbolos por medio de 6 bits [28]. El diagrama de constelaciones para la modulación QAM se muestra en la Figura 2.8(c) y (d) observándose que cada bit está codificado en 2^n estados.

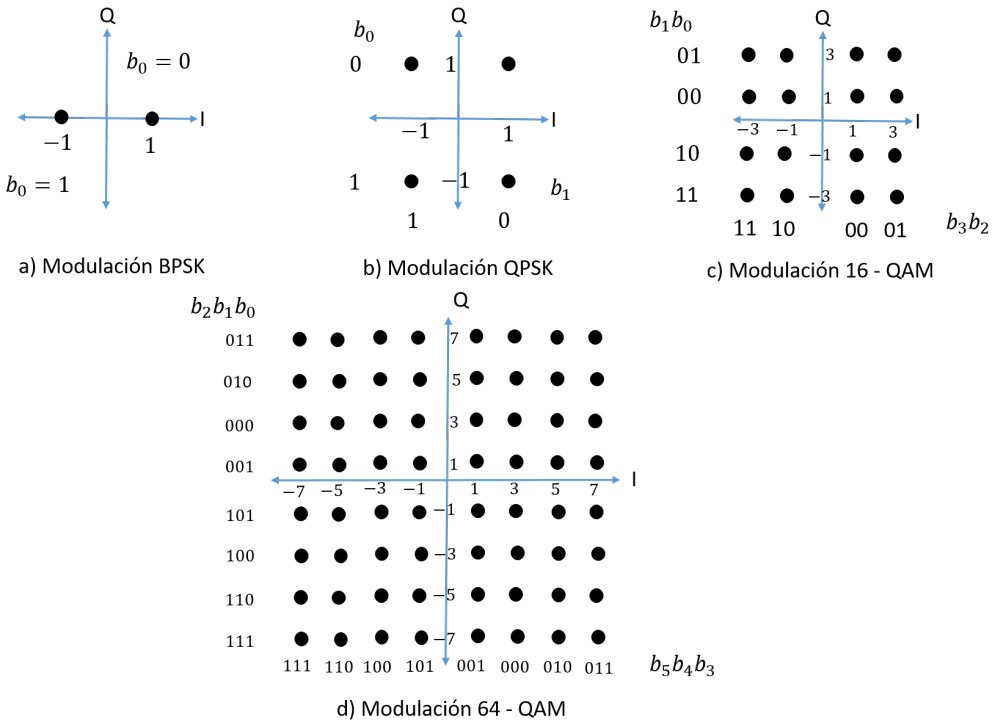


Figura 2.8: Constelación de modulación: BPSK, QPSK, 16-QAM y 64-QAM [25].

La modulación BPSK es la que menor tasa de transmisión tiene. QPSK transmite al doble de velocidad de BPSK. En el caso de la modulación QAM ofrece una mayor tasa de transmisión pero presenta una mayor tasa de error.

2.3.3. Capa de acceso al medio

La capa de control de acceso al medio (MAC) establece los parámetros para el acceso al canal de comunicación. El protocolo MAC IEEE 802.11p garantiza la transmisión de la información y el uso del canal inalámbrico [17]. Además emplea Enhanced Distributed Channel Access (EDCA) en cuatro categorías de acceso junto a una prioridad dependiente del tipo de mensaje y aplicación que se ejecute.

2.3.3.1. Esquema de acceso al canal mejorado

El acceso mejorado al canal distribuido (EDCA) provee la calidad de servicio al estándar IEEE 802.11p, el cual está basado en IEEE 801.22e [29]. EDCA está basado en el acceso múltiple de detección de portadora con prevención de colisión (Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)) utilizando cuatro clases de acceso, Access Class (AC). Las clases de acceso se utilizan para mapear 8 niveles de User priority (UP), prioridad de usuario, de acuerdo a la aplicación empleada [30].

Un nodo que necesita transmitir detectará primero el medio y si este está libre para un Arbitration inter-frame spacing (AIFS) (espacio de arbitraje entre tramas). Todos los AC ejecutan un proceso de espera independiente, *backoff*. El procedimiento de espera se conoce como ventana de *backoff* Bandwith (BW) y funciona de la siguiente manera:

- El nodo a transmitir, inicia con un contador BW en un tiempo de *backoff* al azar dentro del intervalo

$[0, CW - 1]$. Donde el valor inicial de Contention window (CW) (ventana de contención), es igual a CW_{min}.

- Si el contador de *slots* de tiempo llega a 0, se inicia la trasmisión.
- Si el medio está ocupado, el contador se detiene. Si hay varios nodos y su BW llega a cero al mismo instante se produce una colisión.
- El tamaño del intervalo aumentará (doble), cuando el medio esté ocupado y el BW sea 0. Si falla el intento de transmisión posterior, el valor de CW será igual a CW_{max} [31].

Para garantizar que los mensajes de seguridad altamente relevantes se puedan intercambiar de manera oportuna y confiable, incluso cuando se opera en un escenario denso, el protocolo MAC 802.11p tiene en cuenta la prioridad de los mensajes utilizando diferentes clases de acceso. Hay cuatro categorías de tráfico de datos disponibles con diferentes prioridades:

- **BK o AC0** *Background traffic*, en los niveles de prioridad 1 y 2
- **BE o AC1** *Best Effort*, en los niveles de prioridad 0 y 3
- **VI o AC2** *Video*, en los niveles 4 y 5
- **VO o AC3** *Voice*, en los niveles de prioridad más altos, 6 y 7.

En la Tabla 2.3 se muestra los diferentes valores de Arbitration Inter-Frame Space Number (AIFSN) y CW para los diferentes tipos de datos [32].

Tabla 2.3: Parámetros EDCA para aplicaciones en IEEE 802.11p

Tipo de Dato	AC	AIFSN	CW_{min}	CW_{max}
Background	BK	9	15	1023
Best Effort	BE	6	15	1023
Video	VI	3	7	15
Voice	VO	2	3	7

En la Figura 2.9 se muestra un ejemplo en el cual para una estación A, se reducirá en uno el contador de *backoff* siempre que el canal se detecte inactivo durante un *time slot*, se detendrá cuando el canal esté ocupado y continuará después de que el canal se detecte inactivo nuevamente para el periodo AIFS $[AC_i]$. En la estación C, si el canal está ocupado (durante el AIFS $[AC_i]$), el acceso es diferido. La estación continuará monitorizando el canal hasta que esté inactivo durante un tiempo de AIFS $[AC_i]$. Se inicia el procedimiento de *backoff* para minimizar la probabilidad de colisión.

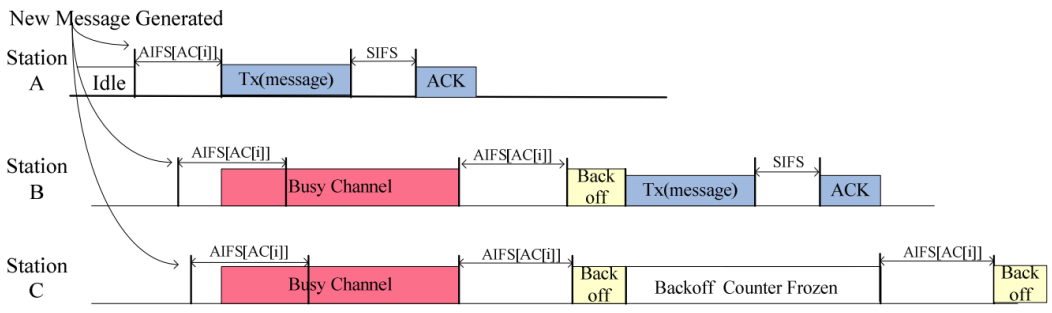


Figura 2.9: Acceso al canal en IEEE802.11p [29]

Cuando el contador de *backoff* disminuye a cero en la estación B, C transmitirá el paquete. Teniendo en cuenta que cada estación tiene cuatro AC, puede ocurrir una colisión cuando simultáneamente más de una cola de AC inicia una transmisión. Un planificador dentro de la estación evitará este tipo de colisión interna al otorgar la cola de AC de mayor prioridad para transmitir los paquetes, mientras que la AC de menor prioridad se ha bloqueado e inició el proceso de *backoff* para volver a intentar la transmisión.

2.3.3.2. CSMA/CA

El estándar IEEE 802.11, utiliza un tipo de protocolo conocido como CSMA/CA. Este algoritmo evita las colisiones en la transmisión, en lugar de descubrir una colisión como lo hace el algoritmo Carrier Sense Multiple Access with Collision Detection (CSMA/CD) [33]. En el sistema CSMA/CA, cuando una estación identifica el fin de una transmisión, espera un tiempo aleatorio antes de transmitir, disminuyendo así la probabilidad de colisión.

2.3.4. Estructura de la capa MAC

La capa MAC está compuesta por un encabezado, el cuerpo de la trama y por el campo Frame Check Sequence (FCS), el cual contiene el código de redundancia cíclica, la estructura de la capa MAC se muestra en la Figura 2.10.

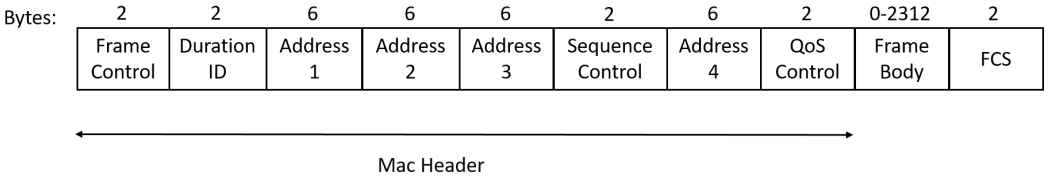


Figura 2.10: Estructura del encabezado MAC

- **`Frame Control`**: Este campo está estructurado como se presenta en la Figura 2.11

Figura 2.11: Estructura del campo *Frame Control* [34]

  - **`Protocol Version:`** con los bits 0 0 se muestra la versión del estándar IEEE 802.11.
  - **`Type:`** Identifica si es una trama de datos, de control o de gestión.
  - **`To DS / From DS:`** Este campo muestra si la trama se está enviando o es recibida del sistema de distribución Distribution Service (DS).
  - **`More Fragments:`** Campo que indica con el bit 1 si el paquete de una capa superior ha sido fragmentado.
  - **`Retry:`** Con el bit 1 se indica si la trama es una retransmisión.
  - **`Power Management:`** Permite activar el modo de ahorro de energía.
  - **`More Data:`** Campo que indica si una estación tiene tramas que enviar hacia una estación o destino específico.

- **`WEP Field`:** Permite conocer si la trama ha sido encriptada con el algoritmo Wireless Equivalent Privacy (WEP).
- **`Order`:** Campo que permite enviar los paquetes en orden [34].

- **`Duration / ID`:** Campo que contiene la duración del periodo reservado para una estación.
- **`Address`:** Los campos de direcciones (`Address 1`, `Address 2`, `Address 3` y `Address 4`) contienen direcciones de 48 bits, especificando lo siguiente:
  - Source Address (SA), dirección de origen.
  - Destiny Address (DA), dirección de destino.
  - Transmission Address (TA), dirección de transmisión.
  - Reception Address (RA), dirección de recepción.
  - Basic Service Set Identifier (BSSID), identificador para el conjunto de servicios básicos.

  Estos campos dependen de los los bits *To DS* y *From DS* en el campo de control de la trama [32]. Esto se muestra en la Tabla 2.4.

Tabla 2.4: Contenido del campo dirección en encabezado MAC

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

- **`Sequence Control`:** En este campo se encuentra el número de secuencia y el número de fragmento de la trama que está siendo enviada.
- **`Frame Body`:** Campo que contiene la carga útil y varía según el tipo de trama que utilice.
- **`FCS`:** Campo que utiliza un código de redundancia cíclica de 32 bits para verificar la integridad de la trama.

2.4. Hardware

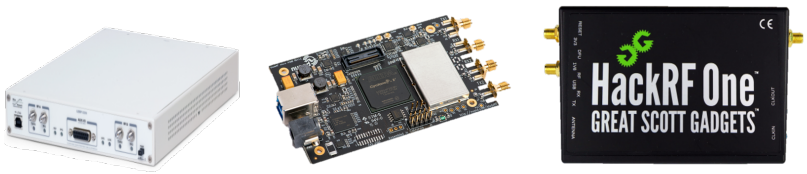
En el presente trabajo de titulación se usan diferentes dispositivos. Para la implementación de la comunicación de los OBUs y los RSUs se usa un dispositivo SDR. En el RSU, además de la comunicación se implementa el hardware para emular un semáforo; en la implementación se utiliza Arduino y un módulo relé. Además se cuenta con dispositivos RF que ayudarán a la implementación de la OBU y de la RSU respectivamente.

2.4.1. SDR

Los dispositivos SDR proveen una plataforma de desarrollo destinada a la investigación. Los equipos SDR tienen etapas Analog to Digital Converter (ADC) y Digital to Analog Converter (DAC) que permiten el muestreo directo de señales. Además, los SDR utilizan amplificadores, mezcladores, y núcleos Phase-Locked Loop (PLL), los cuales permiten realizar prácticas y experimentos en distintas áreas de la ingeniería [35].

Existen varios dispositivos SDR con las características mencionadas, dentro de ellos están: USRP [36], BladeRF [37][38] y HackRF One (Figura 2.12). El BladeRF y HackRF One son plataformas *open hardware*. El más accesible en términos de costo es el HackRF One; este no es vendido directamente por el fabricante

por lo que se puede encontrar en el mercado entre $ 300 y $ 400. Este SDR permite el envío y recepción de datos; está diseñado para la experimentación y desarrollo de soluciones en el ámbito de las comunicaciones inalámbricas. Este dispositivo es capaz de transmitir o recibir señales desde 1 MHz hasta los 6 GHz; además, puede ser programado para su funcionamiento autónomo [39]. El Anexo C.2 amplía la información relacionada con este dispositivo.



a) USRP b) BladeRF c) HackRF One

Figura 2.12: Dispositivos SDR

2.4.2. Módulos

Dentro de los módulos a describir se encuentran Arduino y el módulo relé. El módulo Arduino es una placa electrónica conformada por un microcontrolador re-programable y pines de entrada/salida digital. Mediante los pines se realiza la conexión entre el microcontrolador y sensores y actuadores. Su programación se lo realiza mediante el *software* Integrated Development Environment (IDE) *open source* de Arduino que se lo puede descargar gratuitamente [40, 41]. En la actualidad se cuenta con varios modelos de placas Arduino, cada uno con sus propias características.

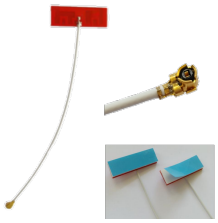
Por otro lado, un módulo relé es un interruptor mecánico conformado por dos estados, normalmente cerrado o normalmente abierto, dejando pasar la corriente o no. Se lo puede controlar mediante voltajes de 5V. La señal de control puede ser enviada desde cualquier microntrolador, en este caso se usa Arduino. Dentro del mercado podemos encontrar módulos con más relés o incluso módulos compuestos de un solo relé. Estos módulos pueden ser alimentados con 3.3V y 5V C.4.

2.4.3. Dispositivos RF

Como se mencionó antes, en el presente trabajo de titulación se utilizan diferentes tipos de dispositivos RF (Figura 2.13): una antena microstrip, una antena ANT 500, un amplificador lineal, un Cable R-178, y varios conectores. continuación se describe cada uno de ellos.

- **Antena microstrip:** Conocidas como antenas *patch*, están conformadas por un parche metálico dispuesto sobre un sustrato dieléctrico. Generalmente tienen la forma rectangular o circular con dimensiones de media longitud de onda ($\lambda/2$). Son versátiles respecto a la frecuencia de resonancia, polarización, patrones de radiación e impedancia. Estas antenas son usadas para el trabajo en frecuencias de 1 hasta 100 GHz (Figura 2.13a) [42, 43].
- **Antena ANT 500:** Es la antena que viene incorporada al HackRF One, fabricada por Great Scott Gadgets está diseñada para funcionar en el rango de 75 MHz hasta 1 GHz. Está construida con acero inoxidable tiene una longitud que varía entre 20 cm a 88 cm. Esta antena tiene una impedancia de 50 Ohmios integrada a un conector SubMiniature version A (SMA) macho con eje giratorio y codo ajustable (Figura 2.13b) [10].

- **Amplificador lineal:** Un amplificador lineal o amplificador de potencia para RF son considerados la última etapa de un sistema de comunicación inalámbrico y se los coloca antes de las antenas. Su función básica es tomar la señal de RF de baja potencia, ya con la codificación, modulación de datos y en la frecuencia deseada y aumenta la intensidad de la señal desde algunos milivatios hasta miles de vatios para su posterior envío (Figura 2.13c) [44].
- **Cable R-178:** Cable coaxial conductor que contiene una impedancia de 50 *Ohm*, trabaja a una frecuencia de hasta 6 GHz, con una atenuación de 4.06 dB/m, este conductor es utilizado en aplicaciones de transmisión de datos, comunicaciones de radiofrecuencia y es utilizado en diversos equipos de baja potencia (Figura 2.13d) [45].
- **Conectores:** Los conectores SMA, son terminales que van incrustados en el borde de la placa o dispositivo por medio de una soldadura a las líneas conductoras del dispositivo, estos conectores están diseñados para un acople a 50 *Ohm* (Figura 2.13e).



(a) Antena microstrip [46]

(b) Antena ANT 500 [10]

(c) Amplificador lineal para RF [47]

(d) Cable R-178 [48]

(e) Conectores SMA [49]

Figura 2.13: Dispositivos de radio frecuencia

2.5. Software

Los dispositivos SDR son equipos programables. Por lo que existen varios tipos de *software* que se utilizan para programar estos dispositivos, dentro de ellos están: Matlab [50], LabView [51] y GNU Radio. De las herramientas mencionadas solo la última es *open source*. GNU Radio proporciona bloques de procesamiento de señales para implementar radios de *software*; y se puede usar con *hardware* de RF externo de bajo costo. GNU Radio permite utilizar SDRs con *hardware* o usando un entorno de simulación. Es ampliamente utilizado en

investigación, industria, academia, gobierno y entornos de aficionados para apoyar al desarrollo de comunicaciones inalámbricas como los sistemas de radio del mundo real [11]. El Anexo C.1 describe con mayor detalle este software.

2.6. Equipos de medición

Estos equipos de medición se utilizarán en pruebas de laboratorio, y permitirán corroborar los procedimientos realizados dentro de las diferentes pruebas. En esta sección se los describe de una manera resumida, mientras que en el Anexo C.5 se detallan más características de estos equipos.

- **Analizador de espectros HP-8593E:** Trabaja en el rango de 9KHz hasta los 6GHz. Este analizador de espectro brinda una gran capacidad de medición para aplicaciones de RF y microondas. Sus modos de medición se combinan con el rendimiento del equipo para proporcionar soluciones personalizadas para su aplicación.
- **Tarjeta para mediciones VNA:** Un Vector Network Analyzer (VNA) o analizador vectorial de redes permite el estudio de las características de señales eléctricas, enfocándose en los parámetros de dispersión y reflexión; permite obtener mediciones de amplitud y fase.
- **Sensor de Potencia NI-USB-5681:** Trabaja en el rango desde los 10 MHz hasta los 18 GHz, con un rango de potencia desde los -40 dBm a +20 dBm. Permite realizar medidas precisas de distintas señales que van desde fuentes de un solo tono a múltiples tonos a formas de onda digitales y complejas de banda ancha.

2.7. Trabajos relacionados

En el desarrollo de las ciudades inteligentes se han elaborado varios proyectos usando redes VANETs [52], a continuación se presenta una breve descripción de las principales propuestas disponibles en la literatura.

En la detección de accidentes de tráfico en [2], proponen el sistema e-NOTIFY que permite el envío de mensajes a un centro de emergencias utilizando las redes vehiculares con la comunicación V2V y V2I. e-NOTIFY pretende mejorar la atención a los afectados post-accidente aumentando la posibilidad de recuperación y supervivencia. Los vehículos de esta red deberán incorporar una OBU, la misma que detectará cuando ha habido un accidente grave para los ocupantes y recopilará información de los sensores en el automóvil. Debido a la versatilidad se puede analizar distintos casos de aplicación. En [53] se estudia la gestión de tráfico, en su análisis resalta el parámetro de la densidad de vehículos. Además realizan pruebas en diferentes escenarios y priorizan el consumo de combustible.

En [54] se propone un algoritmo para gestionar el tráfico en un ambiente simulado, las variables usadas son: número de vehículos con y sin comunicación (tasa de penetración de la tecnología) e intersecciones, obteniendo como resultado de su trabajo, la circulación de automóviles sin generar congestión vehicular. El departamento de transporte de Estados Unidos en [55] analiza varias aplicaciones: información del clima, notificación de infracciones, señalización, además de la comunicación V2I. La información que se obtiene puede servir a un usuario pedestre para alertar de eventos o proveer información.

Utilizando el simulador NS3, en [56] se implementa la capa física del estándar IEEE 802.11p, al utilizar una simulación no se tiene en cuenta la encapsulación y desencapsulación de tramas, codificaciones y los modelos no se basan en efectos como desviación rápida y multitrayectoria. Las implementaciones físicas en *hardware* no se las realizan a menudo por los costos de los equipos. Los autores de [57] presentan como aplicación un

protocolo de comunicación utilizando una OBU entre una entidad de sanción y un vehículo infractor. Utilizando equipos locomate, la RSU envía notificaciones a la OBU informando de la infracción cometida. Se han realizado simulaciones en SUMO, NS2, MOVE y NS3 en donde se integra IEEE 802.11p con WAVE LTE, encontrando valores de retardo, pérdidas en la transmisión y aumentando el tamaño de la red [56].

En el estudio del estándar IEEE 802.11p generalmente se realizan simulaciones debido al costo de equipos y software propietarios para su implementación; sin embargo, al ser simulaciones el comportamiento del sistema se simplifica, y los resultados no toman en cuenta varios parámetros. En una simulación no se puede considerar todos los factores que intervienen en una implementación. Las condiciones ambientales como: temperatura, humedad, entre otras varían a cada instante. Los obstáculos, interferencia o reflexión de señales cambian de forma aleatoria. Los niveles de voltaje, potencia, ruido no son constantes. Al momento de implementar un sistema todos estos factores repercuten en el resultado final. Estos efectos se intentan contrarrestar con diversos métodos matemáticos a lo largo del desarrollo del sistema. Es así que, combinando la funcionalidad de un *hardware* SDR con el procesamiento de señales, en [58] implementa el protocolo IEEE 802.11 a/g/p usando el *hardware* USRP. En el trabajo lograron transmitir con un alcance de hasta 3m y una velocidad máxima de 1.82 Mbit/s. Además, analiza el protocolo a diferentes frecuencias. Los autores de [59] presentan la implementación del protocolo IEEE 802.11p sobre el *hardware* USRP en donde se ha desarrollado un *framework* para la simulación y experimentación del estándar. El proyecto ha trabajado sobre el *software* libre GNU Radio.

2.8. Conclusiones

Como parte de la investigación teórica se observa que el estándar IEEE 802.11p nace con la intención de cubrir las especificaciones de una red ad hoc destinadas a la redes vehiculares. Las redes vehiculares al estar compuestas de nodos fijos o móviles y poseer una arquitectura dinámica, necesitan que la capa PHY adopte un sistema OFDM y se base en el estándar IEEE 802.11a y que la subcapa MAC garantice la transmisión de información mediante un control de acceso inalámbrico.

Mediante GNU Radio se alcanza una versatilidad al programar por medio de bloques o mediante la creación de módulos programados en Python o en C++. Además, las herramientas de visualización permiten observar los resultados de diferentes maneras y con la posibilidad de ejecuciones jerárquicas. Una de las características que más resaltan es el montaje del programa implementado en GNU Radio sobre *hardware* como lo son HackRF, RTL-SDR, USRP, entre otros.

Los dispositivos SDR dependiendo de sus características permiten recibir y transmitir señales a diferentes frecuencias, utilizando diversos anchos de banda y con diversas resoluciones en sus dispositivos ADC. Dentro de los dispositivos SDR, HackRF One es de lo más versátiles y económicos, cuenta con una antena telescópica para la transmisión o recepción, además es posible conectar otro tipo de antena gracias al conector SMA hembra.

Para comprobar la programación y validar los resultados con parámetros específicos, los equipos de medición descritos en este capítulo permitirán analizar el espectro y verificar parámetros como la frecuencia a la que se está trabajando, el ancho de banda, y el nivel de potencia de las señales de salida. Además, con el uso del equipo VNA se verificará parámetros de las antenas y amplificadores como: parámetros S, frecuencia de resonancia, entre otros.

CAPÍTULO 3

Análisis y diseño de la aplicación VANET

En el presente capítulo se expondrá los programas y dispositivos empleados en el desarrollo de este trabajo de titulación. En primera instancia se analiza el *software* y el *hardware* empleado. Se expone la arquitectura para el desarrollo de la aplicación VANET que se implementará. Se detalla el sistema operativo y las librerías que se utilizan. Se muestra el montaje de los equipos para el transmisor y el receptor. Finalmente se presenta la aplicación a diseñarse, las conexiones entre componentes y la interfaz a utilizar.

3.1. Análisis de software y hardware

Los dispositivos SDR son equipos programables. Existen varios paquetes de *software* que se utilizan para programar estos dispositivos (ver Sección 2.5). Matlab y Labview son programas propietarios y necesitan de versiones actuales y licencias específicas para trabajar con los equipos SDR. Por factores de licencias y sobre todo por disponibilidad de librerías, se utilizará GNU Radio en la versión: 3.7.11. Este *software* es de código abierto, utiliza programación por bloques y permite implementar programas enfocados en el tratamiento de señales. GNU Radio se instala de preferencia sobre un sistema operativo Linux, posee una interfaz gráfica y cuenta con dos opciones de programación: módulos y bloques. Estos módulos y/o bloques pueden ser propios del *software* o de usuarios que colaboran con este. Un módulo o bloque puede ser programado en Python o C++. El uso de variables, bloques, módulos y demás características son presentadas en el Anexo C.1.

En el mercado existen varios equipos SDR. En la Tabla 3.1 se muestra un resumen de los equipos más populares con sus respectivas características [60].

Tabla 3.1: Dispositivos SDR

Dispositivo	Frecuencia Min	Frecuencia Max	Ancho de Banda	RX	TX	Resolución ADC
RTL-SDR	24 MHz	1766 MHz	3.2 MHz	Si	No	8
Funcube Pro	64 MHz	1700 MHz	0.096 MHz	Si	No	16
Funcube Pro+	410 MHz	2050 MHz	0.192 MHz	Si	No	16
blade RF	300 MHz	3800 MHz	40 MHz	Si	Si	12
MatchStiq	300 MHz	3800 MHz	28 MHz	Si	Si	12
Hack RF	10 MHz	6000 MHz	20 MHz	Si	Si	12
USRP	10 MHz	6000 MHz	64 MHz	Si	Si	12

En este trabajo se busca que el dispositivo SDR tenga la capacidad de transmisión y recepción. Existen varios equipos con estas características. La Universidad de Cuenca cuenta con los dispositivos HackRF ONE.

El rango de frecuencia del HackRF ONE está dentro del que se establece para el estándar IEEE 802.11p. Este dispositivo trabaja en diferentes niveles de potencia. Cuando está en modo recepción se recomienda no exceder de -20 dBm. La potencia a la que trabaja en modo transmisión depende directamente de la frecuencia a la que se esté utilizando [61], como lo indica la Tabla 3.2. Las características y especificaciones del dispositivo HackRF ONE se encuentran en el Anexo C.2.

Tabla 3.2: Relación entre frecuencia y la potencia de transmisión de HackRF ONE

FRECUENCIA	POTENCIA
10 MHz - 2150 MHz	15 dBm - 5 dBm.
2150 MHz - 4000 MHz	5 dBm - 0 dBm.
4000 MHz - 6000 MHz	0 dBm - -10 dBm.

3.2. Arquitectura

En la Figura 3.1 se presenta la arquitectura para la aplicación VANET a implementar. El sistema consta de dos partes. Por un lado se tiene la RSU, la cual transmitirá la información y por otro lado se encuentra la OBU, esta tiene la misión de recibir y procesar la señal recibida.

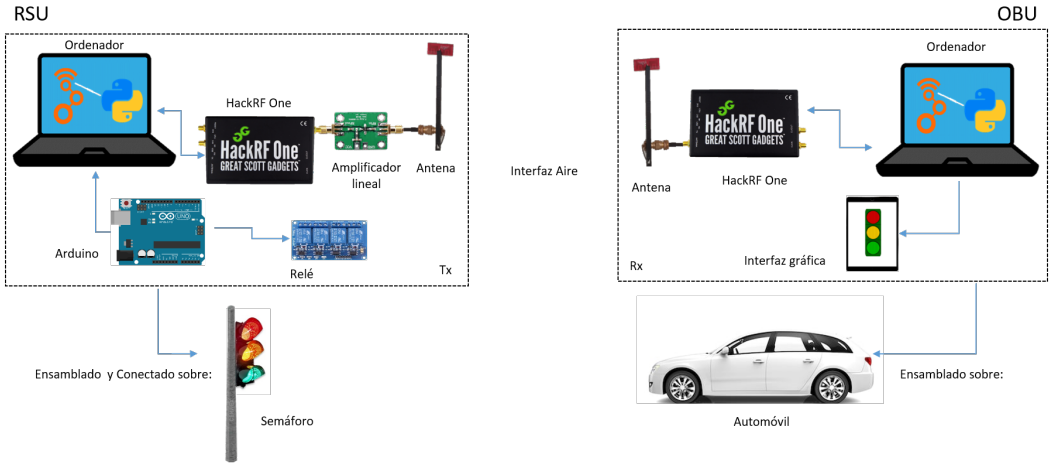


Figura 3.1: Arquitectura para la aplicación VANET

La RSU está compuesta de dos partes. La primera parte consiste en obtener la información. La información viene dada por el estado de un semáforo. Este semáforo es controlado de forma independiente con un Arduino y un módulo relé. Estos dispositivos permiten al semáforo cambiar su estado, es decir, encender los paneles de luces en una secuencia indicada y por un tiempo definido. El estado del semáforo es enviado por puerto serial hacia un archivo `txt` guardado en el computador.

La segunda parte consiste en enviar la información hacia el receptor. La información es obtenida del archivo `txt` utilizando código Python y el *software* GNU Radio, donde se encuentra implementado el estándar IEEE 802.11p para el transmisor. El programa está implementado en función de la estructura del estándar: capa MAC y capa PHY. Cabe mencionar que el programa en GNU Radio se lo debe cargar previamente al HackRF ONE.

El paquete de información bajo el estándar 802.11p es construido por medio del HackRF ONE y se encuentra listo para ser enviado hacia el amplificador lineal. El amplificador lineal aumenta el nivel de potencia de la señal, y esta es emitida a través de una antena hacia la OBU.

La OBU es la encargada de receptar la señal. Esta conformada por un ordenador, el dispositivo HackRF ONE, y la antena receptora. El proceso comienza con la adquisición de la señal enviada desde la RSU, para esto se utiliza la antena que está conectada al dispositivo HackRF ONE y este al ordenador. Dentro del ordenador se encuentra el *software* correspondiente al estándar IEEE 802.11p definido para la recepción. El programa procesará la señal recibida para obtener los datos utilizando la capa PHY y la capa MAC. Una vez obtenida la información, el estado del semáforo se presenta al usuario mediante una interfaz gráfica en una ventana del navegador web. Todos los elementos que conforman la OBU, y que permiten obtener la información enviada desde la RSU, están integrados dentro de un automóvil.

3.3. Parámetros de software

Para el funcionamiento del transmisor y receptor es necesario contar con un sistema operativo basado en Linux por lo que se ha utilizado el *software* de virtualización VirtualBox en donde se crearon dos máquinas virtuales con las mismas características. La Tabla 3.3 presenta las características de las máquinas virtuales.

Tabla 3.3: Características de las máquinas virtuales creadas en VirtualBox para el transmisor y receptor

Maquina Virtual	
Sistema Operativo	Ubuntu 18.04 LTS (64 bits)
Memoria RAM	8 GB
Número de Procesadores	1
Video Memory	16 MB
Graphics Controller	VBoxVGA
Network Adapter	NAT
USB Controller	USB 3.0 Controller

Una vez instalado Ubuntu 18.04 en las máquinas virtuales se instalan las herramientas necesarias para el transmisor y receptor. En el Anexo A.1 se detalla los comandos para la instalación de GNU Radio.

Los bloques del transmisor y receptor están basados en un repositorio de código abierto realizado por Bastian Bloessl, el mismo que se lo puede clonar desde [62]. En este repositorio se implementa el módulo DSRC dedicado a redes vehiculares bajo el estándar IEEE 802.11p. En este módulo se encuentran dos repositorios adicionales, *gr-ieee802-11* una implementación basada en GNU Radio del estándar IEEE 802.11p, y *gr-foo*, en donde se halla el bloque de Wireshark. Los bloques para GNU Radio de estos dos repositorios son instalados con una secuencia de comandos que se enlistan en el Anexo A.2. Las herramientas `make`[63] y `cmake` son usadas para la construcción de los bloques. En este trabajo fue usada la versión 3.7.2 de `cmake`. El proceso de instalación de estas herramientas se encuentra detallado en el Anexo A.2.

3.4. Disposición de los equipos

3.4.1. Antenas

En la OBU, la antena se coloca en el capó del vehículo. El vehículo se debe movilizar a diferentes velocidades. En la RSU la antena se encuentra en la parte superior del semáforo. En ambas posiciones las antenas están susceptibles al movimiento que es provocado por el viento.

La estructura que se muestra en la Figura 3.2 protege a las antenas permitiendo que se mantengan en una posición fija y evitando que estas se caigan o deterioren el cable de conexión. Se dispone de dos estructuras, estas difieren en el ancho de la muesca que se encuentra en la base. La muesca permite que se fijen en el vehículo y en el semáforo. La estructura diseñada para las antenas tiene un mecanismo que permite inclinar en diferentes grados a las antenas. El mecanismo de inclinación se logra por medio de un eje sujeto a la base de la estructura. La inclinación debe ser tal que la señal enviada obtenga la mayor distancia con línea de vista desde el transmisor (RSU) hacia el receptor (OBU).

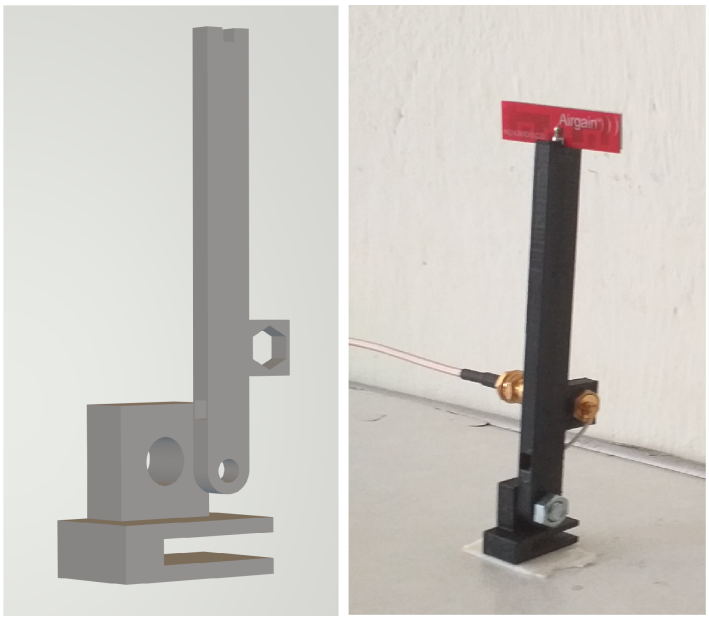


Figura 3.2: Estructura para la antena

3.4.2. Receptor

En cuanto al receptor, un vehículo tendrá a bordo el dispositivo HackRF ONE, el cual va conectado a un computador como se muestra en la Figura 3.3. La OBU receptará todas las señales enviadas desde el transmisor.

Figura 3.3: Montaje de la OBU

3.4.3. Transmisor

La RSU estará colocada en un semáforo, dentro de la estructura del semáforo se encuentran los siguientes elementos:

- HackRF ONE
- Amplificador RF
- Arduino Uno
- Módulo relé
- Cable RF y cables de conexión

El semáforo cuenta con un módulo relé, el cual es controlado por un Arduino Uno. El Arduino activa el módulo relé y controla los tres paneles de luces. Cada panel cuenta con una matriz de 126 leds, de color rojo, amarillo y verde, con un total de 378 leds y es alimentado a 120 V. El semáforo que se muestra en la Figura 3.4 está basado en las características e imposiciones que aplican las leyes de tránsito en el país. Tiene 26 cm de ancho, 76 cm de alto y 33 cm de profundidad. Además, está diseñado con la protección necesaria para ser instalado en ambientes externos.

En la parte interna del semáforo existe el espacio suficiente para colocar todo el hardware necesario para instalar el transmisor del estándar IEEE 802.11p. En la Figura 3.4 se muestra el hardware instalado dentro del semáforo. El semáforo contiene una salida para la alimentación a 120V, la conexión del HackRF ONE y la conexión del Arduino al ordenador.

Figura 3.4: Montaje de la RSU

3.5. Aplicación vehicular

La aplicación vehicular permite el envío del estado de un semáforo (RSU) y la recepción del estado por parte de una OBU utilizando el estándar IEEE 802.11p. La Figura 3.5 muestra las conexiones entre los diferentes dispositivos que conforman la RSU. Cada panel de luces, es conectado a un módulo relé y este a un Arduino Uno. En el Arduino se establece la secuencia de encendido y apagado de luces y el tiempo en que cambiará de estado. El estado del semáforo está ligado a un valor que por medio del puerto serial es enviado a la máquina virtual, esta a su vez pasa el valor como un parámetro al bloque encargado de captar la información y colocarlo como dato a transmitir.

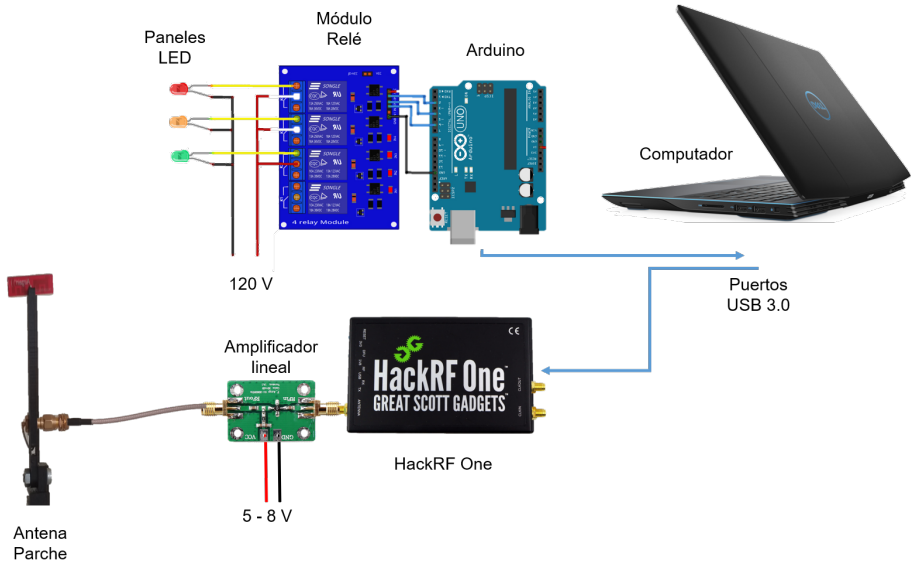


Figura 3.5: Conexiones para la RSU

El paquete codificado utilizando el estándar IEEE 802.11p llega al OBU. El paquete se desencapsula capa por capa hasta llegar a la carga útil. En la carga útil se encuentra el estado que fue enviado por el semáforo. Un programa desarrollado en Node-red mostrará, en una interfaz gráfica, el estado del semáforo. De esta forma el conductor conocerá el estado del semáforo sin tener que estar viendo directamente a este. La interfaz es mostrada en la Figura 3.6 y el código utilizado para toda la aplicación se detalla en el Anexo B.

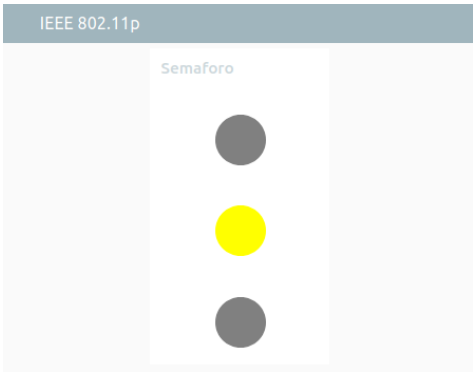


Figura 3.6: Interfaz gráfica

3.6. Conclusiones

El *software* GNU Radio permite trabajar con dispositivos SDR. El *hardware* usado en el presente proyecto es el HackRF ONE. El control del HackRF ONE se lo realiza usando librerías desarrolladas por la comunidad de software libre alrededor del mundo.

Dentro de la arquitectura planteada, se contempla un amplificador lineal en el transmisor para tener más potencia en la señal de salida. Del lado de la OBU no existe un amplificador dado que solo es necesario receptar la señal.

El estándar que se implementa no impone el estado del semáforo ya que la información referente al estado del semáforo es enviada desde el Arduino por medio de un archivo externo.

CAPÍTULO 4

Implementación de la aplicación VANET

En este capítulo se desarrolla la implementación de los diferentes bloques de la aplicación VANET en GNU Radio. La implementación comprende la capa física y la capa de enlace de datos del estándar IEEE 802.11p (Figura 4.1). Como se mencionó en el Capítulo 3 la RSU tendrá la capacidad de transmitir datos, mientras que la OBU recibirá los datos y los procesará. Para ello tanto el transmisor como el receptor tendrán cargados en su *hardware* un programa específico desarrollado en GNU Radio que cumple con las características del estándar IEEE 802.11p.

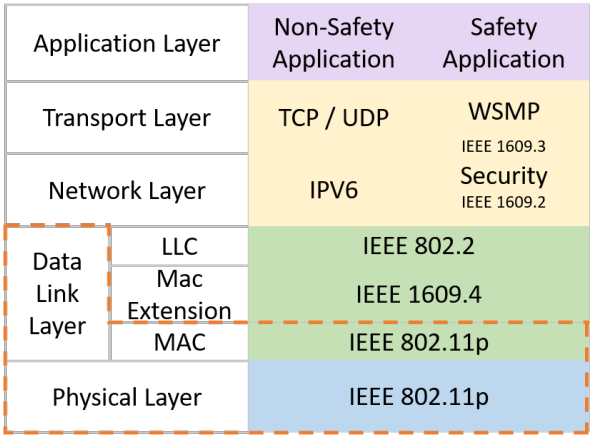


Figura 4.1: Capas utilizadas en el estándar IEEE 802.11p

4.1. Transmisor

Para la implementación se utilizará dos métodos de compilación de los programas, la compilación normal con la visualización de resultados con una interfaz QT Graphical User Interface (GUI) y la compilación jerárquica que está basada en usar dos o más archivos, uno dependiente de otro. Por medio de la programación de bloques se ha considerado dos programas: transmisor y receptor. Estos tendrán el programa de la subcapa MAC y el programa para la capa PHY que será ejecutado de manera jerárquica (Sección C.1.3.3). El programa del

transmisor se muestra en la Figura 4.2.

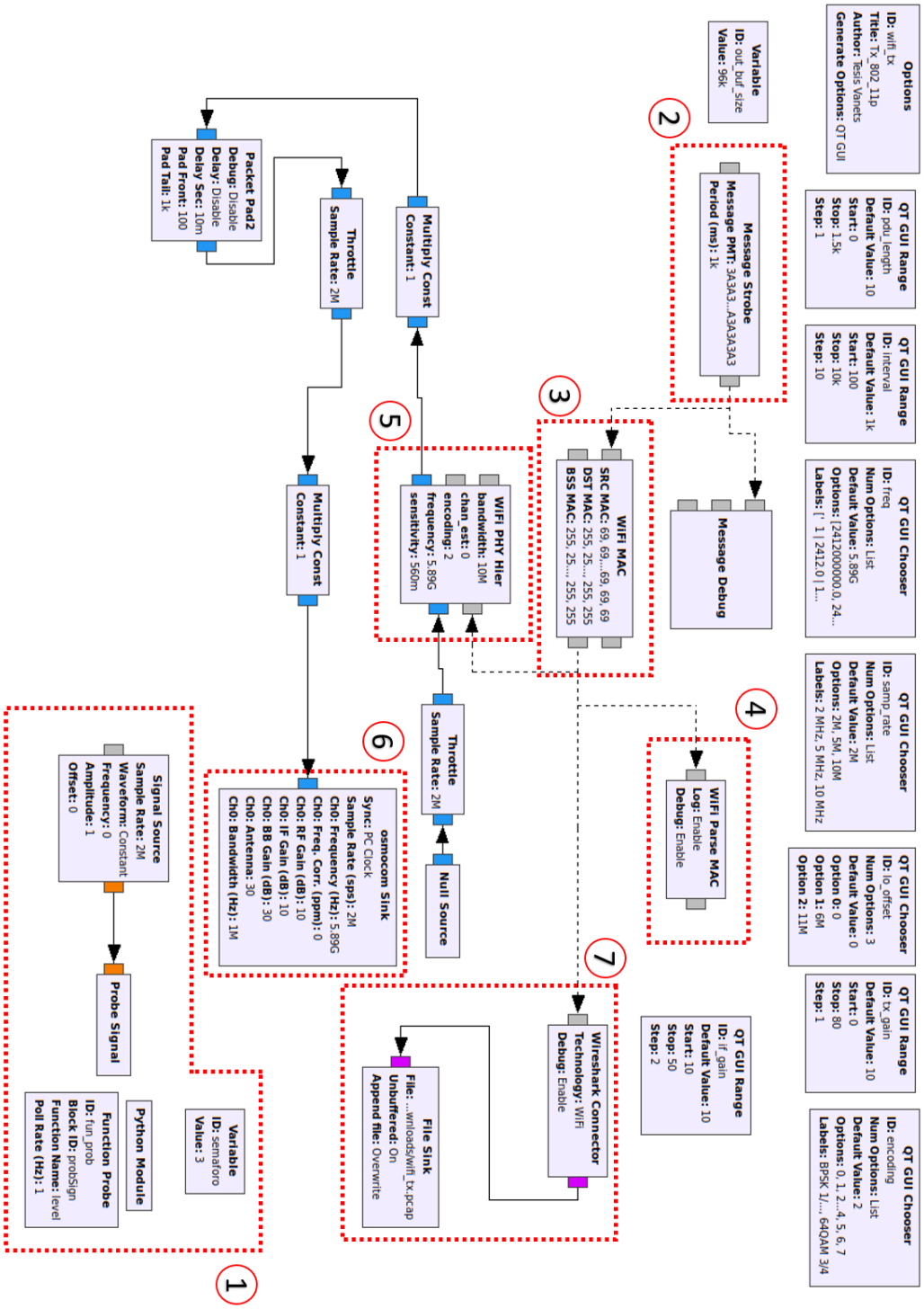


Figura 4.2: Implementación del Transmisor en GNU Radio

El programa implementado en GNU Radio (Figura 4.2) describe el funcionamiento del programa para el transmisor. En el cual se ha enumerado las diferentes etapas: lectura del dato a enviar, encapsulación de los datos

en la capa MAC y PHY, y transmisión de la señal OFDM por parte del HackRF ONE.

1. En la sección 1 de la Figura 4.2 se encuentran tres bloques implementados en GNU Radio. El bloque *Python Module* lee un archivo `txt` en donde está almacenado un valor entre 1 y 3 que hacen referencia a los tres estados que puede tener el semáforo. El bloque *Python Module* envía el estado hacia el bloque *Variable(Semaforo)* y este a su vez lo enviará al bloque *Message Strobe* ubicado en la sección 2. La configuración de bloques que intervienen en esta sección se muestran en la Figura 4.3.

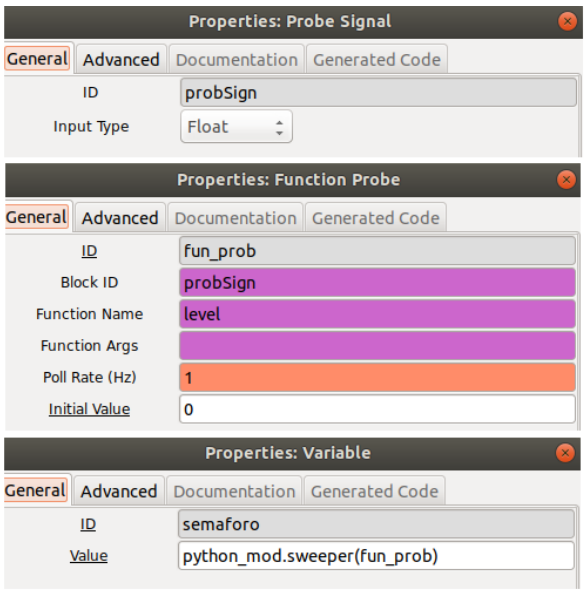


Figura 4.3: Lectura de datos externos

2. La información adquirida por el bloque *Message Strobe* (sección 2 de la Figura 4.2) pasa a ser una señal constante. Este bloque envía mensajes en intervalos de tiempo. El intervalo es configurado por medio de la variable *interval*. En la Figura 4.4 se muestra la configuración del bloque *Message Strobe*.

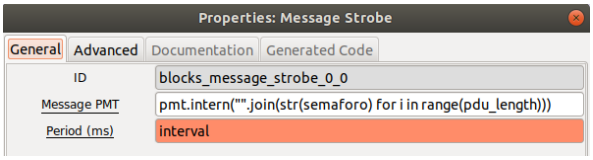


Figura 4.4: Bloque *Message Strobe*

3. Desde el bloque *Message Strobe* el mensaje se envía al bloque *Wifi Mac* (sección 3 de la Figura 4.2). En este bloque se ingresan las direcciones MAC: fuente, destino y el conjunto básico de servicios. Cada dirección es un identificador único de red. Las direcciones están compuestas de 48 bits, se utilizan 6 grupos de números hexadecimales. Por medio de las letras A, B, C, D, E y F se denotan los números 10, 11, 12, 13, 14 y 15. El número total de posibles direcciones MAC que podrían existir es 2 a la potencia 48. De esta manera, cada nodo en una red sabe cuándo el tráfico está destinado para sí. La configuración del bloque *Wifi Mac* se presentan en la Figura 4.5.

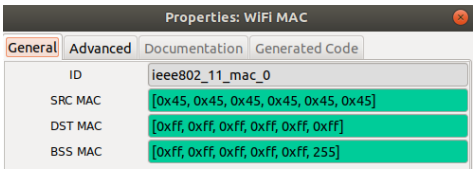


Figura 4.5: Bloque *Wifi Mac*

Para poder transmitir en *broadcast*, la dirección MAC debe ser `FF:FF:FF:FF:FF:FF`. Las direcciones MAC que se utilizan se muestran en formato hexadecimal y son las siguientes:

- SRC MAC: `45:45:45:45:45:45`
- DST MAC: `FF:FF:FF:FF:FF:FF`
- BSS MAC: `FF:FF:FF:FF:FF:FF`

El bloque *Wifi Mac* valida el ingreso de una dirección MAC correcta. Luego, este bloque determina la longitud del mensaje a transmitir con un máximo de 1500 bytes. Además, en cada transmisión que se realiza va aumentando en 1 un número de secuencia el cual empieza en 0.
El encabezado MAC contiene el campo de control de *frame* (con un valor de 8), la duración del *frame* y el número de secuencia. El encabezado tiene un tamaño de 24 bytes, al cual se le agrega 4 bits más para la verificación de secuencia de *frame*.
El encabezado MAC se encapsula dentro del MAC Service Data Unit (MSDU). De igual forma el MSDU se encapsula dentro del Physical Service Data Unit (PSDU). Finalmente, se calcula el CRC32 que es una función que permite identificar errores generados entre datos de origen y destino.

4. Con los *frames* formados, la salida del bloque *Wifi Mac* se conecta al bloque *Parse Mac*, correspondiente a la sección 4 de la Figura 4.2. Este bloque divide en secciones fáciles de procesar, valida la composición de los paquetes y añade etiquetas, las cuales analizan la correcta sintaxis. La configuración de este bloque se presenta en la Figura 4.6.

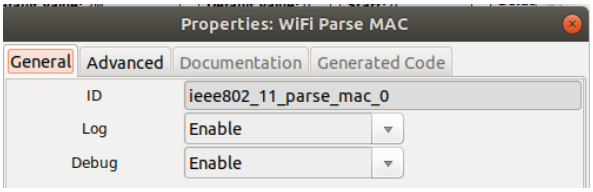


Figura 4.6: Bloque *Parse Mac*

5. El bloque *Wifi Mac* se conecta con el bloque de la capa PHY, representado mediante el nombre de *WiFi PHY Hier* (sección 5 de la Figura 4.2) y su configuración se muestra en la Figura 4.7. Este bloque contiene los siguientes parámetros: ancho de banda, frecuencia de operación, estimación de canal, tipo de modulación y sensibilidad.

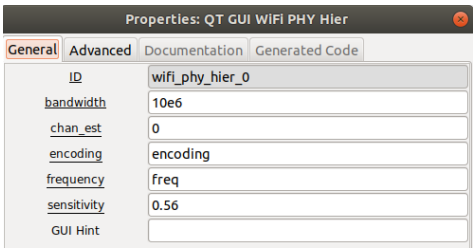


Figura 4.7: Bloque *WiFi PHY Hier*

Se utiliza un selector para ingresar los valores de ancho de banda que pueden ser: 2, 5 y 10 MHz, de acuerdo a lo que define el estándar IEEE 802.11p. La frecuencia de operación está enlazada al canal de frecuencia. En la Tabla 4.1 se presentan los posibles canales para el estándar IEEE 802.11a/g/n/p.

Tabla 4.1: Número de canal y frecuencia respectiva para el estándar IEEE 802.11 a/g/n/p

CHH	F(GHz)	CHH	F(GHz)	CHH	F(GHz)	CHH	F(GHz)	CHH	F(GHz)
1	2.412	2	2.417	3	2.422	4	2.427	5	2.432
6	2.437	7	2.442	8	2.447	9	2.452	10	2.457
11	2.462	12	2.467	13	2.472	14	2.484	34	5.170
36	5.180	38	5.190	40	5.200	42	5.210	44	5.220
46	5.230	48	5.240	50	5.250	52	5.260	54	5.270
56	5.280	58	5.290	60	5.300	62	5.310	64	5.320
100	5.500	102	5.510	104	5.520	106	5.530	108	5.540
110	5.550	112	5.560	114	5.570	116	5.580	118	5.590
120	5.600	122	5.610	124	5.620	126	5.630	128	5.640
132	5.660	134	5.670	136	5.680	138	5.690	140	5.700
142	5.710	144	5.720	149	5.745	151	5.755	153	5.765
155	5.775	157	5.785	159	5.795	161	5.805	165	5.825
172	5.860	174	5.870	176	5.880	178	5.890	180	5.900
182	5.910	184	5.920						

Los canales utilizados en el estándar IEEE 802.11p van desde el 172 hasta el 184 [4]. Debido a que se utiliza el canal de control, se coloca por defecto el canal 178 correspondiente a la frecuencia de 5.890 GHz.

El bloque *WiFi PHY Hier* permite enlazar el programa de la capa PHY (Figura 4.8) con el de la capa MAC. Dado que está configurado de manera jerárquica, permite tener programas por separado.

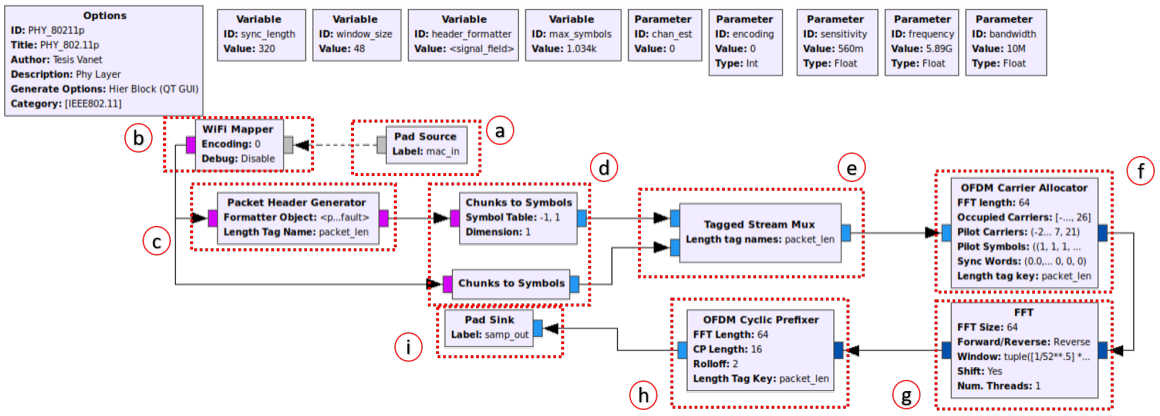


Figura 4.8: Capa física IEEE 802.11p en Tx

El programa de la capa PHY para el transmisor se muestra en la Figura 4.8 y cuenta con las siguientes etapas.

a) Mediante el bloque *Pad Source* se obtiene la información de la capa MAC (sección a en la Figura 4.8). Bajo la etiqueta `mac_in` toda la estructura de la capa MAC formada y validada es ocupada en la capa PHY. En la Figura 4.9 se muestra la configuración del bloque *Pad Source*.

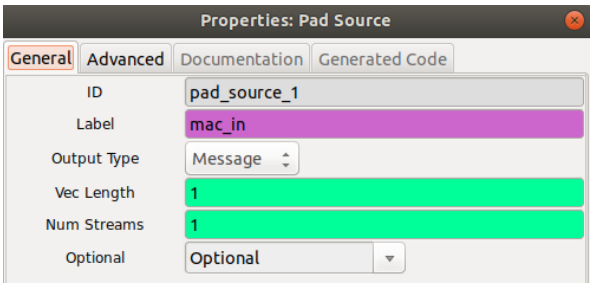


Figura 4.9: Bloque *Pad Source*

b) El bloque *Pad Source* está conectado al *WiFi Mapper*. Este bloque se integra con el parámetro de codificación. Dentro del bloque *WiFi Mapper* (Figura 4.10) se realiza el proceso de aleatorización, entrelazado y codificación. Para el proceso de aleatorización se utiliza el polinomio generador $x^7 + x^4 + 1$.

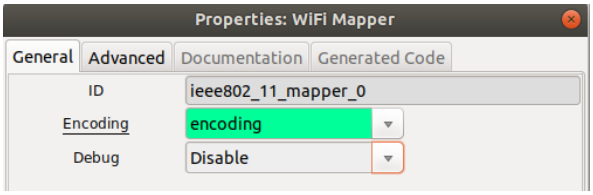


Figura 4.10: Bloque *WiFi Mapper*

Por defecto se configura una modulación BPSK para el encabezado mientras que, para la carga útil o los datos, se puede elegir las siguientes modulaciones: BPSK, QPSK, 16 QAM y 64 QAM. Se utiliza un factor de normalización para cada tipo de modulación de la siguiente manera:

- modulación BPSK: factor = 1
- modulación QPSK: factor = $1/\sqrt{2}$
- modulación 16 QAM: factor = $1/\sqrt{10}$
- modulación 16 QAM: factor = $1/\sqrt{42}$

Este proceso se realiza utilizando la codificación Gray. En la Tabla 2.2 se encuentran las características de cada tipo de modulación utilizando un ancho de banda de 10 MHz.

c) Para la asignación del encabezado se usa el bloque *Packet Header Generator* mostrado en la sección c de la Figura 4.2. La configuración del bloque *Packet Header Generator* se presenta en la Figura 4.11.

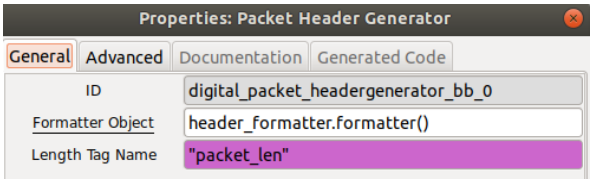


Figura 4.11: Bloque *Paquet Header Generator*

d) La señal obtenida se considera como la señal en serie. Se agrupa la señal en paralelo en *n* números de bits de acuerdo al tipo de modulación que se utiliza. Mediante el bloque *Chuncks to Symbols* (sección d en la Figura 4.8) se realiza la correspondiente modulación, es decir, se le asigna un valor complejo (fase y cuadratura) a cada uno de los diferentes estados. En la Figura 4.12 se muestra la configuración del bloque *Chuncks to Symbols* para cada uno de los tipos de bits (*payload* y *headers*).

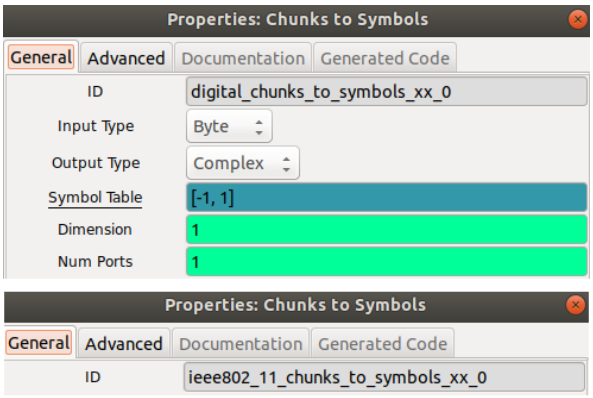


Figura 4.12: Bloques *Chunks to Symbols*

e) Los bits modulados son nuevamente estructurados en un solo *frame*, para ello se multiplexa los bits *headers* con los *payloads* mediante el bloque *Tagged Stream Mux* (sección e en la Figura 4.8). Los campos a configurar en este bloque se muestran en la Figura 4.13.

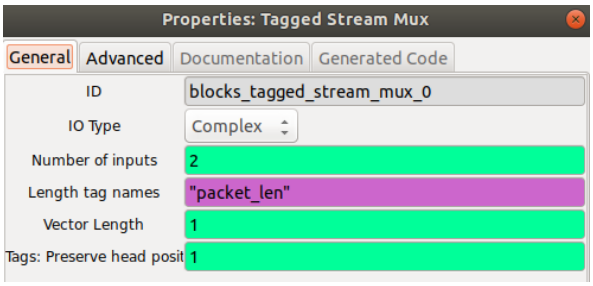


Figura 4.13: Bloque *Tagged Stream Mux*

f) Los *frames* multiplexados son enviadas al bloque *OFDM Carrier Allocator*. Este bloque, mostrado en la sección f en la Figura 4.8, permite mapear cada uno de los bits modulados en una portadora de datos. En la Figura 4.14 se encuentra la configuración de este bloque. El bloque asigna 52 portadoras, de las cuales 48 corresponden a portadoras de datos y 4 corresponden a portadoras piloto. Las palabras de sincronización se utilizan para estimar correctamente la señal captada por el receptor.

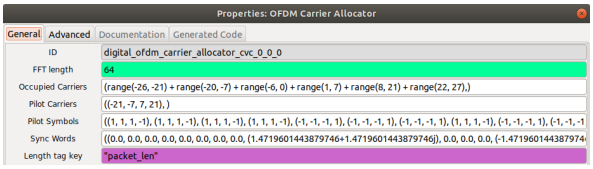


Figura 4.14: Bloque OFDM Carrier Allocator

g) Las señales son enviadas al bloque *FFT* (sección g en la Figura 4.8). En este bloque se realiza la trasformada de Fourier. Para la transformada se utiliza una longitud 64 sub-portadoras debido al número de sub-portadoras que se tiene. La Figura 4.15 muestra la configuración del bloque Fast Fourier Transform (FFT).

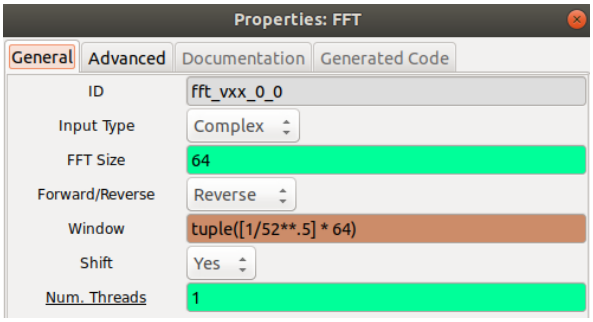


Figura 4.15: Bloque FFT

h) Luego de realizar la transformada inversa de Fourier, se asigna un prefijo cíclico por medio del bloque *OFDM Cyclic Prefixier* mostrado en la sección h de la Figura 4.8. Dado que el tamaño de la FFT es de 64, la longitud del prefijo cíclico es de 16. De las 64 sub-portadoras, 48 sub-portadoras son de datos, 4 sub-portadoras son piloto, 1 sub-portadora DC y 11 sub-portadoras son vacías. En la Figura 4.16 se muestra la configuración del bloque *OFDM Cyclic Prefixier*.

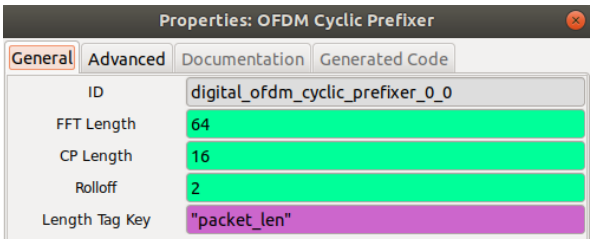


Figura 4.16: Bloque OFDM Cyclic Prefixer

i) Toda la señal OFDM se direcciona al bloque *Pad Slink* (sección i en la Figura 4.8) para regresar al programa principal (Figura 4.17), recalcando que la etiqueta a utilizar en este bloque es `samp\_out`.

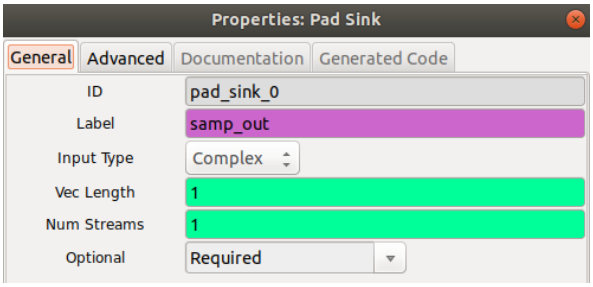


Figura 4.17: Bloque Pad Slink

6. En el programa principal se emplea un bloque de multiplicación para aumentar la amplitud de la señal. Esta señal ingresa al bloque *Osmocom Slink* (sección 6 de la Figura 4.2) el cual se encarga de transmitir toda la señal formada a través del dispositivo HackRF ONE. La configuración de este bloque se encuentra en la Figura 4.18.

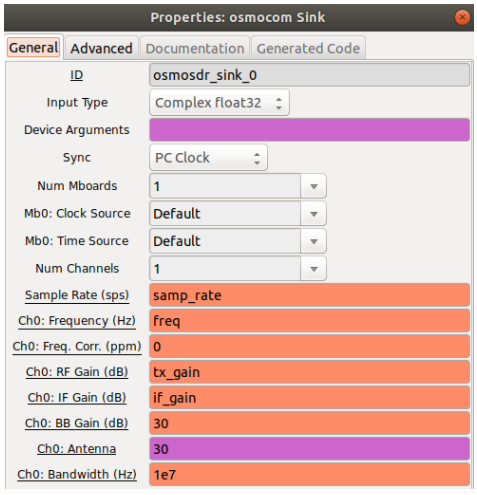


Figura 4.18: Bloque Osmocom Slink

7. Para analizar posteriormente el paquete se coloca el bloque *Wireshark Connector*. La captura de paquetes que ofrece este bloque se puede guardar en un archivo externo usando el bloque *File Slink*. Estos bloques se encuentran en la sección 7 de la Figura 4.2.

Al momento de ejecutar el programa, se presenta una interfaz de usuario como se presenta en la Figura 4.19. En esta interfaz se selecciona las diferentes variables y selectores, entre estos parámetros están:

- Frecuencia de operación
- Ancho de banda
- Esquema de modulación
- Potencia de transmisión en el HackRF ONE

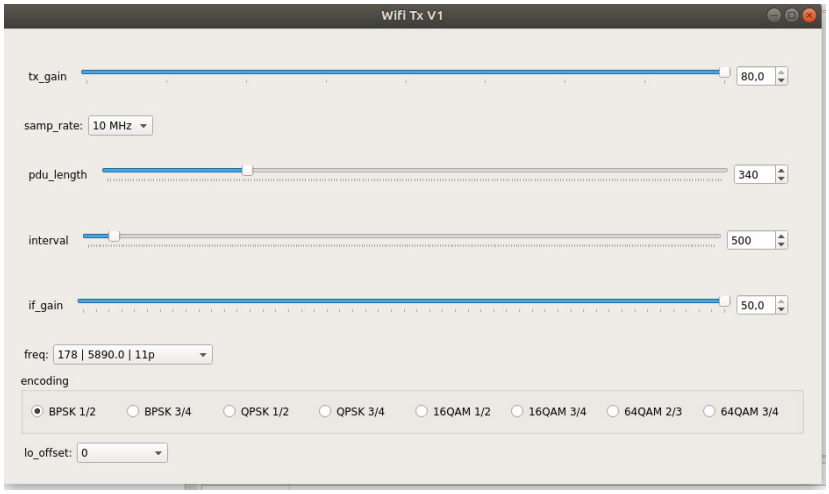


Figura 4.19: Interfaz de usuario del programa transmisor

La Tabla 4.2 presenta varios parámetros que se definen para el estándar IEEE 802.11p.

Tabla 4.2: Características de implementación del estándar IEEE 802.11p

Parámetro	Valor
Ancho de Banda	10
Tasa de transmisión	3, 4.5, 6, 9, 12, 18, 24, 27
Modulación	BPSK, QPSK, 16 QAM, 64 QAM
Tasa de Codificación	1/2, 2/3, 3/4
Sub-portadoras	52
Sub-portadoras de datos	48
Sub-portadoras piloto	4
Transformada de Fourier	64
Espaciamiento entre sub-portadora	0.15625 MHz
Periodo FFT	6.4 us
Periodo FFT	6.4 us
Tiempo de Guarda	1.6
Duración de símbolo	8 us
Duración de secuencia corta	16 us
Duración de secuencia larga	16 us
Duración de preámbulo	32 us
Codificación de error de código	k = 7

4.2. Receptor

Una vez que la señal ha atravesado el medio de comunicación, en este caso el aire, del lado del receptor se obtiene el paquete OFDM mostrado en la Figura 4.20. En el paquete se tiene 12 símbolos para el campo sincronización, este campo es conocido como el preámbulo PLCP. Luego del campo sincronización se encuentra el campo señal y por último el campo MAC.

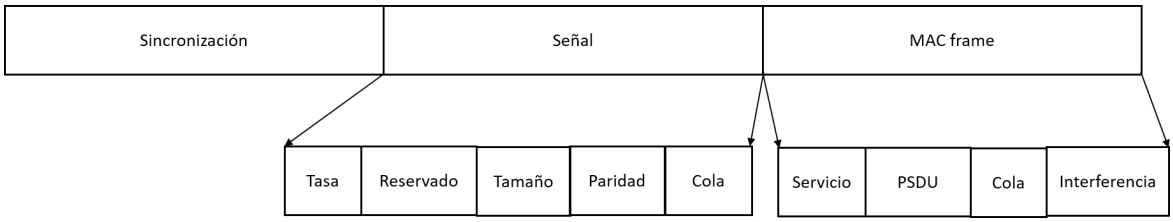


Figura 4.20: Paquete OFDM

Los 12 símbolos OFDM del preámbulo tienen una duración de 32 microsegundos. El preámbulo está dividido en dos partes, como se observa en la Figura 4.21. En la primera parte se utiliza diez símbolos de entrenamiento OFDM con una duración de 1.6 microsegundos. Estos símbolos se envían a través de 12 sub-portadoras. La segunda parte comienza con un tiempo de guarda, que dura 3.2 microsegundos y dos símbolos de entrenamiento OFDM que duran 6.4 microsegundos. Estos dos símbolos utilizan las 52 sub-portadoras disponibles y son moduladas en BPSK.

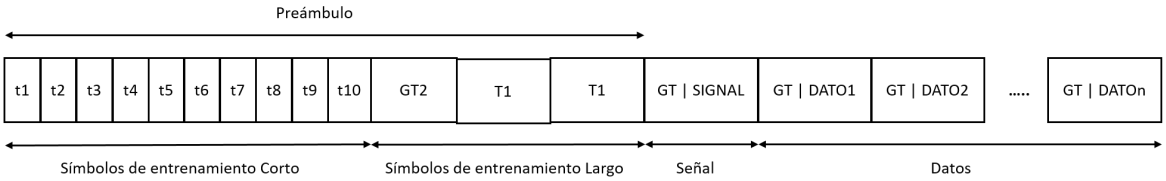


Figura 4.21: Campo PLCP

En el campo señal se tiene los subcampos: tasa de datos, longitud del *frame*, *parity* y *tail*. El esquema de codificación utilizado depende de la tasa de transmisión. Los campos `tail` y `pad` del campo de datos se utilizan de manera que la longitud total sea un múltiplo entero de la longitud de un bloque. Además, la longitud de un bloque depende de la modulación y del esquema de codificación.

La Figura 4.22 presenta el diagrama de bloques del receptor. En el receptor se utiliza un solo programa que contempla la capa física y la capa de enlace de datos. Este programa contiene la detección del *frame* y el proceso de decodificación de la señal. Este proceso del programa del receptor se explica a continuación.

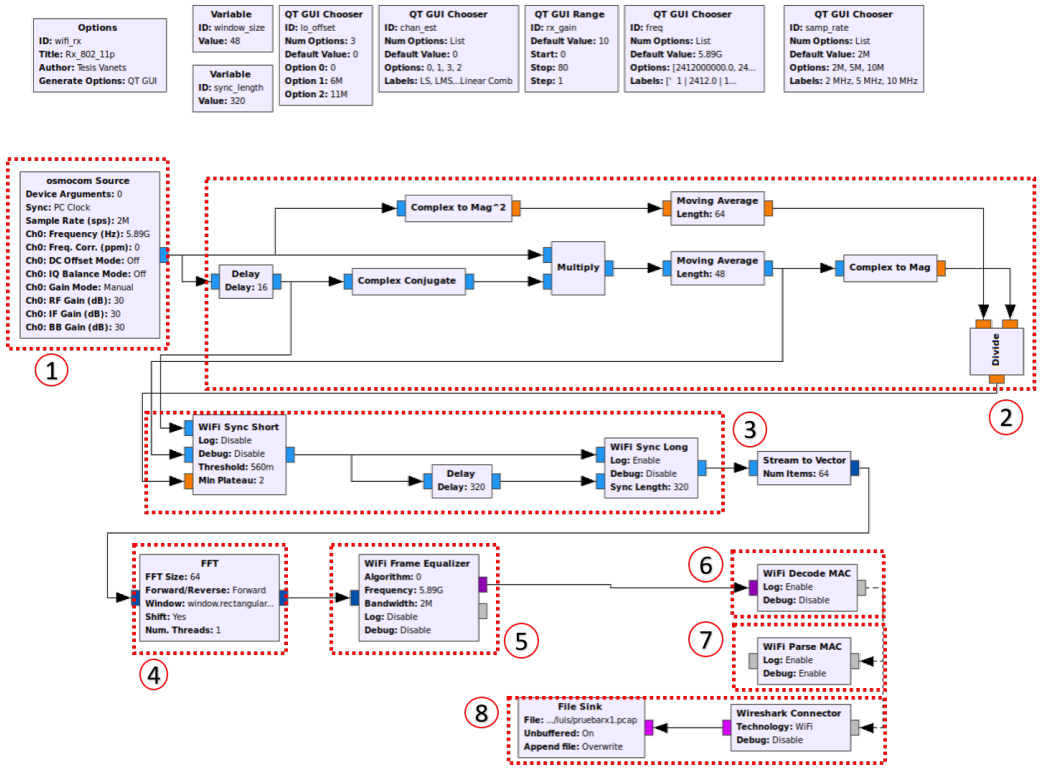


Figura 4.22: Implementación del receptor en GNU Radio

1. El proceso empieza con el bloque *Osmocom Source* (sección 1 de la Figura 4.22). Este bloque utiliza el dispositivo HackRF ONE y obtiene la señal del medio de acuerdo a los parámetros establecidos. La configuración de este bloque se presenta en la Figura 4.23.

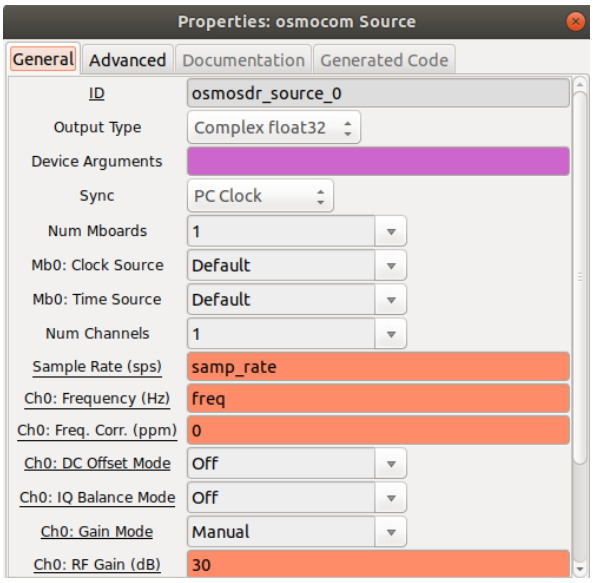


Figura 4.23: Bloque *Osmocom Source*

2. En la sección 2 de la Figura 4.22 se encuentran los bloques necesarios para la detección de *frame*, el programa se basa en el algoritmo de autocorrelación [64]. La secuencia contiene 16 muestras que se repiten 10 veces.
   En la Ecuación (4.1) se calcula el valor de autocorrelación del flujo de muestras entrantes con un retardo de 16, y se suman los coeficientes de autocorrelación en una ventana ajustable. La suma sobre esta ventana se convierte en un promedio móvil que actúa como un filtro de paso bajo. La autocorrelación es alta al comienzo de cada *frame* debido a la propiedad cíclica de la secuencia de entrenamiento corta [65]. Este procedimiento está basado en el algoritmo desarrollado en [66].

$$a(n) = \sum_{k=0}^{N_{win-1}} s(n+k)\overline{s}(n+k+16) \tag{4.1}$$

   Donde:

   a = Valor de autocorrelación
   s = Muestra entrante
   $\overline{s}$ = Complejo conjugado de s
   N_{win-1} = Ventana autoajustable

   Utilizando $N_{win-1} = 48$ para un mejor rendimiento [66], se normaliza la autocorrelación con la potencia media, Ecuación (4.2).

$$p(n) = \sum_{k=0}^{N_{win-1}} s(n+k)\overline{s}(n+k) \tag{4.2}$$

   Donde:

   p = Potencia media

   Para obtener el nivel absoluto de las muestras entrantes, se calcula la norma de la autocorrelación con la potencia media y se define el coeficiente de autocorrelación, Ecuación (4.3). Esto permite al receptor ser independiente del nivel absoluto de las muestras entrantes [64].

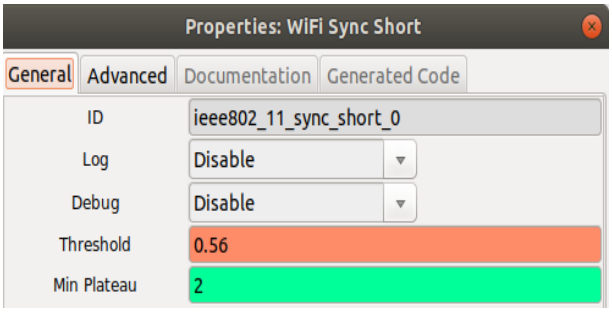
$$c(n) = \frac{|a(n)|}{p(n)} \tag{4.3}$$

Donde:

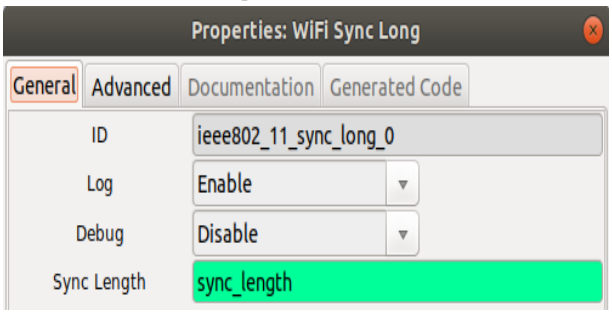
$|a(n)|$ = Magnitud de a

c = Coeficiente de autocorrelación

3. El *frame* detectado pasa al bloque *Wifi Sync Short* (sección 3 de la Figura 4.22). La señal se convierte en muestras y se obtiene el coeficiente de autocorrelación normalizado. En el bloque *Wifi Sync Long* (sección 3 de la Figura 4.22) se corrige el desplazamiento en frecuencia. El coeficiente de autorrelación es utilizado para la alineación de los símbolos. Es importante que tanto el transmisor como el receptor tengan una sincronización previa. Como se muestra en los bloques Osmocom (Figura 4.23), la sincronización viene dada con el reloj de la computadora. Estos bloques se muestran en la Figura 4.24.



(a) Bloque *Wifi Sync Short*



(b) Bloque *Wifi Sync Long*

Figura 4.24: Bloques *Sync Short* y *Sync Long*

Dado que el transmisor y el receptor pueden presentar una ligera variación en sus osciladores se debe contrarrestar el desplazamiento de frecuencias. Considerando $s(n)$ como la muestra del preámbulo corto y $s(n+16)$ como las muestras correspondientes a la propiedad cíclica. La corrección del desplazamiento en frecuencia se puede expresar con la Ecuación (4.4).

$$df = \frac{1}{16} arg\left\{\sum_{n=0}^{N_{short}-1-16} s(n)\overline{s}(n+16)\right\} \tag{4.4}$$

Donde:

df = Desplazamiento de frecuencia

N_{short} = Longitud de las muestras del entrenamiento corto

c = Coeficiente de autocorrelación

Para estimar el desplazamiento en frecuencia se utiliza las 5 últimas secuencias de entrenamiento corto, cada una está formada de 16 secuencias repetidas periódicamente [64]. A cada muestra $s[n]$ le debería corresponder una muestra $s[n+16]$ debido a debido a su propiedad cíclica. No obstante, al introducir ruido en la señal, no se consideraría $s[n]s[n+16]$. Si se desprecia el ruido, el desplazamiento de frecuencia se puede expresar como el argumento de $s(n)\bar{s}(n+16)$ que corresponde a 16 veces la rotación entre cada muestra. Para estimar el valor final, se calcula un promedio (Ecuación (4.4)) y finalmente el desplazamiento de frecuencia se aplica a cada muestra [65].

4. A los símbolos y los preámbulos sincronizados se aplica la transformada de Fourier (sección 4 de la Figura 4.22). La configuración del bloque *FFT* se presenta en la Figura 4.25.

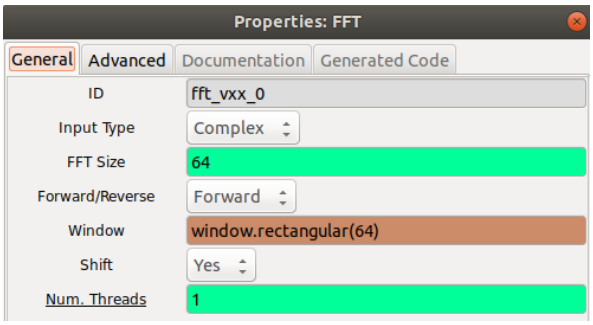


Figura 4.25: Bloque FFT

5. Por medio del bloque de símbolos de ecualización *WiFi Frame Equalizer* (sección 5 en la Figura 4.22), se procede a la corrección del desfase y la estimación del canal. Con la ayuda de las sub-portadoras piloto (4 en total), se corrige la fase y magnitud. En la Figura 4.26 se presenta la configuración de este bloque.

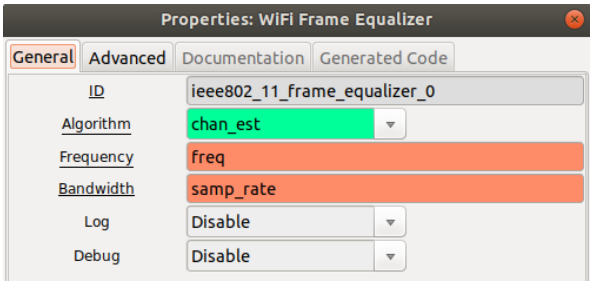


Figura 4.26: Bloque Wifi Frame Equalizer

6. En el bloque *Wifi Decode MAC* que se muestra en la Figura 4.27, se obtiene las secuencias de entrenamiento cortas y largas que contienen el campo de señal, lo cual es un símbolo OFDM modulado en BPSK 1/2. Para la decodificación del código convolucional, la señal de decodificación OFDM coloca una etiqueta, que lleva una tupla de codificación y longitud de *frame*.

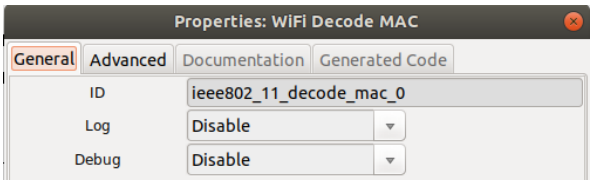


Figura 4.27: Bloque *Wifi Decode MAC*

Para obtener la carga útil se sigue el siguiente proceso: reordenación de los bits, demodulación de los datos, elimina la aleatorización y se emplea un decodificador convolucional.
El bloque recibe vectores con los puntos de constelación en el plano complejo, correspondientes a las 48 sub-portadoras de datos por símbolo OFDM. Según el esquema de modulación utilizado, estas constelaciones se mapean a valores flotantes que representan los bits de la modulación empleada. Dependiendo del esquema de codificación y modulación, los bits de un símbolo se permutan. La permutación es la misma para todos los símbolos de un *frame*.
Para la decodificación convolucional se utiliza un decodificador Viterbi. En el codificador, el estado inicial del aleatorizador se establece en un valor pseudoaleatorio. Como el aleatorizador se implementa con un registro de desplazamiento de retroalimentación de siete bits, son posibles $2^7 = 128$ estados iniciales. Los primeros 7 bits de la carga útil son parte del campo de servicio y siempre se ponen a cero, para permitir que el receptor deduzca el estado inicial del aleatorizador. El mapeo de estos primeros bits al estado inicial se implementa mediante una tabla de búsqueda.
Por último se comprueba si el *frame* está libre de errores. Para ello se calcula el campo de verificación de *frame* y se compara con el campo del paquete recibido, si los campos son iguales, no existen errores.

7. El bloque *WiFi Parse MAC*, al igual que en el diagrama del transmisor, permite validar la composición de los paquetes y añade etiquetas utilizadas para el análisis de una sintaxis correcta.
8. Para analizar posteriormente el paquete se coloca el bloque *Wireshark Connector*. La captura de paquetes que ofrece este bloque se puede guardar en un archivo externo usando el bloque *File Slink*. Estos bloques se encuentran en la sección 8 de la Figura 4.22.

Una vez ejecutado el programa, se inicia la interfaz de usuario mostrada en la Figura 4.28. En esta interfaz se puede seleccionar las diferentes variables y selectores, entre los parámetros a seleccionar están:

- Frecuencia de operación
- Ancho de banda
- Método de estimación de canal
- Potencia de recepción en HackRF ONE

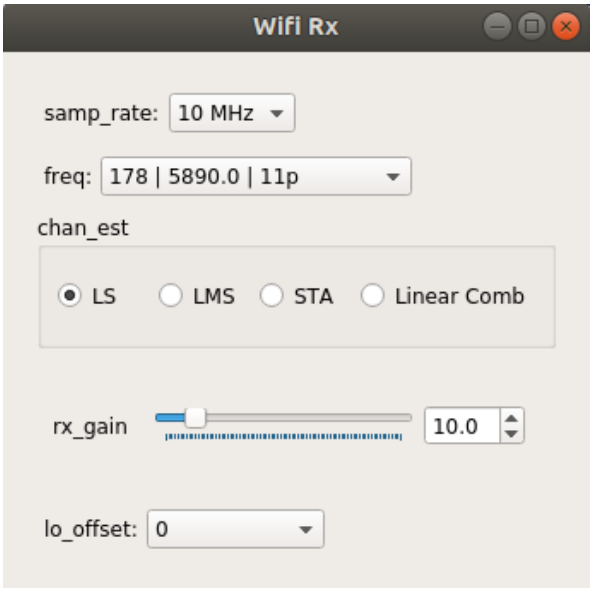


Figura 4.28: Interfaz de usuario del programa receptor

4.3. Análisis de paquetes

En el diagrama del transmisor y en el receptor se generan archivos `.pcap`. Las capturas del tráfico de red se emplean para revisar la estructura de los paquetes. El archivo `.pcap` desde el transmisor se muestra en la Figura 4.29. En primera instancia se observa de forma general el número de paquete, el tiempo de captura, la fuente del archivo, el destino, el protocolo utilizado y la longitud del archivo. En el campo `frame` está el número de bytes que contiene el archivo. En la captura se evidencia que la encapsulación es de tipo IEEE 802.11; además, se muestra el tiempo en el que se generó el paquete, el cual tiene el formato: mes, día, años, hora, minuto, segundo, milisegundo, zona horaria; y el número y la longitud del *frame* que se está analizando.

Radiotap Header es una herramienta para obtener información adicional sobre el *frame*. Esta herramienta es específica para el estándar IEEE 802.11. Dentro de los datos más relevantes mostrados por la herramienta se encuentra el *data rate*, el canal y la frecuencia; además provee información sobre las banderas, los valores de potencia de la antena, ruido de la antena y número de antenas.

En el campo *802.11 radio information* se observa el *data rate* (3 Mb/s), el canal (178 [5.890 GHz]), y los valores de potencia. El campo IEEE 802.11 *Data*, contiene los datos del *frame* de control, entre ellos se muestra la versión, el tipo o subtipo. En este caso el tipo de *frame* es de datos. En las banderas se observa que el *frame* es modo ad hoc, por lo que presentan los siguientes valores: *To DS*: 0 y *From DS*: 0.

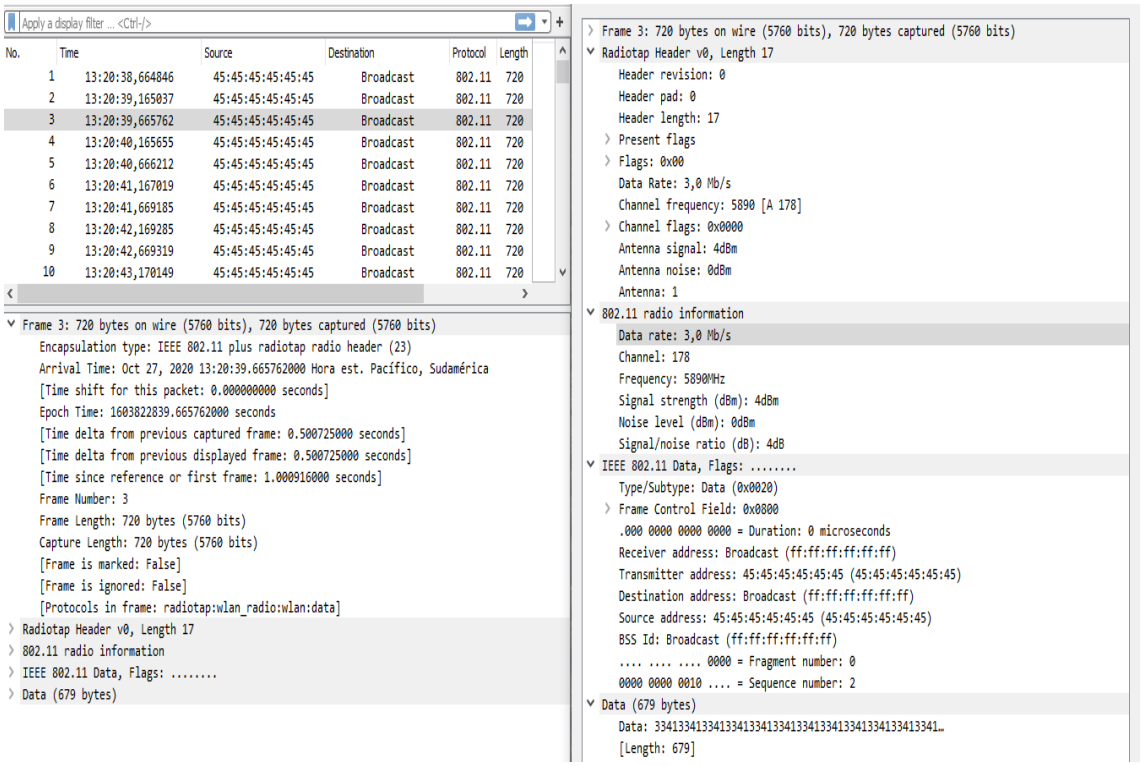
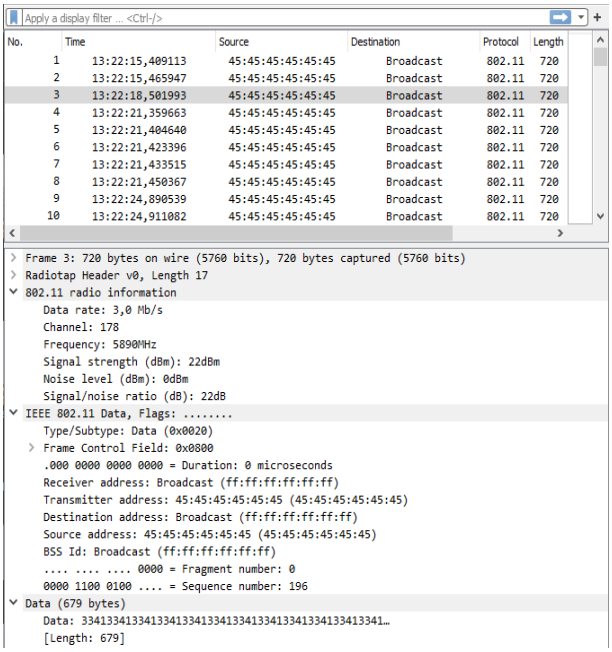


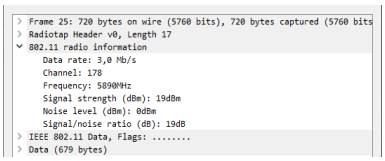
Figura 4.29: Análisis de paquetes en el Transmisor

Dentro de este mismo campo se encuentra la dirección del receptor (`FF:FF:FF:FF:FF:FF`), la dirección del transmisor (`45:45:45:45:45:45`), la dirección de destino (`FF:FF:FF:FF:FF:FF`), la dirección de origen (`45:45:45:45:45:45`) y el Id BSS (`FF:FF:FF:FF:FF:FF`). Se muestra además el número de secuencia del paquete. Por último, se presenta el campo de los datos, en donde se encuentra toda la información útil para el receptor en formato decimal.

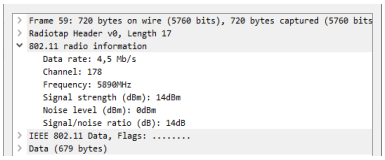
Por otro lado, en el receptor, la información es similar a la del archivo del transmisor. En la Figura 4.30, se encuentra la información de la capa de control de acceso al medio y la capa física. De igual manera, en un inicio se muestra el número de paquete recibido, la hora de recepción, dirección de fuente y destino, protocolo y longitud del archivo.



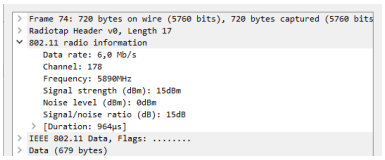
(a) Archivo pcap Receptor



(b) Modulación BPSK 1/2



(c) Modulación BPSK 2/3



(d) Modulación QPSK 1/2

Figura 4.30: Análisis de paquetes en el Receptor

El valor de *data rate* contenido en el campo 802.11 *radio information* es de 3 Mb/s. Esto significa que se está modulando usando BPSK con una tasa 1/2 (Figura 4.30b). Cuando se utiliza la modulación BPSK a una tasa 2/3 el *data rate* es de 4.5 Mb/s (Figura 4.30c). En la Figura 4.30d se muestra el campo *radio information* cuando se utiliza QPSK con tasa 1/2. En el campo *radio information* además se cuenta con información sobre el canal que se está utilizando (178 [5.890 GHz]), los niveles de señal y la duración del *frame*.

En el campo *IEEE* 802.11 *Data* se observa el tipo (en este caso es Datos), el campo de control de *frame*, las direcciones correspondientes al receptor, transmisor, fuente y BSS Id, y al final de este campo está el número de secuencia (196). El campo *Data*, contiene la información enviada por el transmisor que va a ser utilizada para una aplicación definida.

4.4. Conclusiones

Con el programa desarrollado para el transmisor se logra la implementación de diferentes modulaciones desde BPSK hasta QAM, se puede elegir un ancho de banda desde 2 MHz hasta 10 MHz. Además, se añade bloques al programa (sección 1 y sección 2 de la Figura 4.2) con el objetivo de poder interactuar con archivos externos y transmitir el mensaje deseado.

El uso del bloque *WiFi PHY Hier* permite diseñar el programa de la capa PHY en un archivo diferente al del programa de la capa MAC, para poder ejecutarlo de manera jerárquica. Al elaborar el programa de transmisión de esta manera, se identifican las diferentes etapas que forman la capa MAC (Figura 4.2) y la capa PHY (Figura

4.8) y por las cuales la señal OFDM debe atravesar para su encapsulación (sección 3 a sección 5 de la Figura 4.2) y posterior envío (sección 6 de la Figura 4.2).

En cuanto al receptor, su programa para la capa MAC y PHY se desarrolla y ejecuta en un solo archivo. Con ciertas modificaciones en su programa se consigue la detección del *frame*, la estimación del canal, demodulación y decodificación de la señal en recepción, además se ordenan los bits y se recupera de la carga útil.

En recepción, existen pérdidas debido a que el tamaño del *frame* se limita al número de símbolos OFDM. No existe un control en el transmisor dado que no está implementado Request to Send and Clear to Send (RTS/CTS). Esta implementación no es necesaria, salvo el caso en que el dispositivo SDR actúe como transceptor, característica que no posee el HackRF ONE. Además, el receptor exige más procesamiento que el transmisor, debido a los procesos que se debe ejecutar para obtener la carga útil. Están integrados 4 métodos de estimación de canal, sin embargo se utiliza solo el algoritmo de mínimos cuadrados. Este método está basado en los símbolos de entrenamiento.

Cabe mencionar que para un correcto funcionamiento de los dos programas implementados, es necesario la instalación de repositorios y de librerías con versiones específicas. En el Anexo A se encuentra el procedimiento de instalación de software, repositorios y librerías.

A través de los archivos `pcap` se evidencia la estructura de la capa de acceso al medio y la capa física, y se corrobora el cumplimiento de los diferentes parámetros que establece el estándar. Además, por medio de estos archivos se puede analizar: retardo en paquetes, retardo promedio general, pérdida de paquetes y *throughput* normalizado por cada transmisión que se realice.

Capítulo 5

Caracterización del dispositivo HackRF One

El objetivo de este capítulo es caracterizar el dispositivo HackRF ONE en transmisión y recepción. Este dispositivo se usará para la implementación del estándar IEEE 802.11p. Para ello en la parte de transmisión, se desarrolla un programa en GNU Radio que envía una portadora simple. Los factores a analizar son el nivel de potencia de salida y su comportamiento en función de la frecuencia establecida por el estándar. En cuanto a la caracterización del HackRF ONE en recepción, lo que se busca es establecer niveles mínimos y máximos de potencia en donde el dispositivo trabaje de manera confiable.

Además, con el programa desarrollado en GNU Radio para el estándar IEEE 802.11p, se obtendrá los mejores parámetros en frecuencia, potencia, ancho de banda y esquema de modulación a utilizar en las pruebas de campo. Para la realización de estos experimentos se utilizarán varios equipos de laboratorio: analizador de espectro, sensor de medidor de potencia y VNA.

5.1. HackRF ONE en transmisión

La caracterización de este dispositivo se lo realiza mediante la medición de su potencia de transmisión. El objetivo es encontrar en que banda de frecuencia el HackRF ONE transmite a una mayor potencia e identificar el comportamiento del HackRF ONE en la banda de frecuencia en la que está definido el estándar IEEE 802.11p.

5.1.1. Escenario implementado en transmisión

Para la caracterización se implementa el escenario de la Figura 5.1. En el escenario se definen varias etapas: transmisión de una portadora, amplificación de la portadora por parte del amplificador lineal y captura de la potencia mediante el sensor y su software.

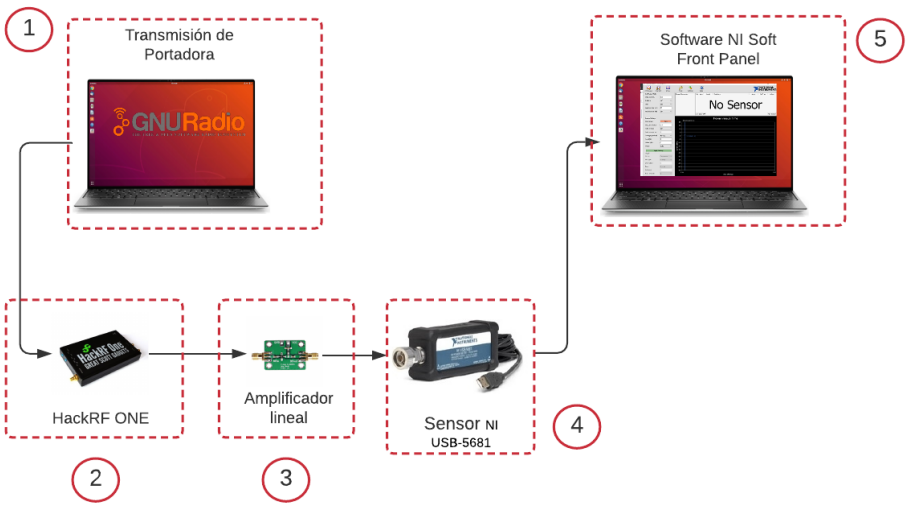


Figura 5.1: Escenario implementado para la medición de potencia de transmisión del HackRF ONE

1. En la sección 1 de la Figura 5.1 se encuentra el ordenador ejecutando un programa en GNU Radio para la transmisión de una portadora. El programa hará un barrido de frecuencia desde 1 GHz hasta los 5.9 GHz. En el bloque de lectura de GNU Radio para el HackRF ONE (*RTL-SDR Source*) se cuenta con una variable denominada *if Gain*. Esta variable actúa directamente sobre el amplificador interno del HackRF ONE, por lo que al colocar un valor de 50, de acuerdo a mediciones realizadas en el laboratorio, se obtiene una ganancia aproximada de 10 dB adicionales.
2. Conectado al ordenador está el HackRF ONE (sección 2 de la Figura 5.1). Se cuenta con tres dispositivos que serán intercambiados para conocer su potencia de transmisión y observar cual presenta un mejor rendimiento. Los tres dispositivos son etiquetados de la siguiente manera:
   - HackRF ONE 1 como `H1_As`
   - HackRF ONE 2 como `H1_Av`
   - HackRF ONE 3 como `H1_Uc`
3. A la salida del HackRF ONE está conectado el amplificador lineal (sección 3 de la Figura 5.1). Este dispositivo tiene como objetivo amplificar la potencia de la señal de salida del HackRF ONE. Al incrementar el voltaje de alimentación del amplificador lineal, su ganancia aumenta. Las características teóricas de este amplificador son las siguientes:
   - Ganancia: 20 dB.
   - Frecuencia de funcionamiento: 20-3000 MHz.
   - Voltaje de alimentación V_{in}: 5 a 9V en corriente directa.

   Los valores teóricos mencionados anteriormente tiene cierta incertidumbre. Por lo que para medir la ganancia real del amplificador lineal se emplea el equipo VNA.
   En la Figura 5.2 se observa la ganancia (parámetro S_{21}) que tiene el amplificador lineal que es de alrededor de 10 dB cuando el voltaje de alimentación es de 7.5V en una ventana de frecuencia de 2.2-3 GHz.

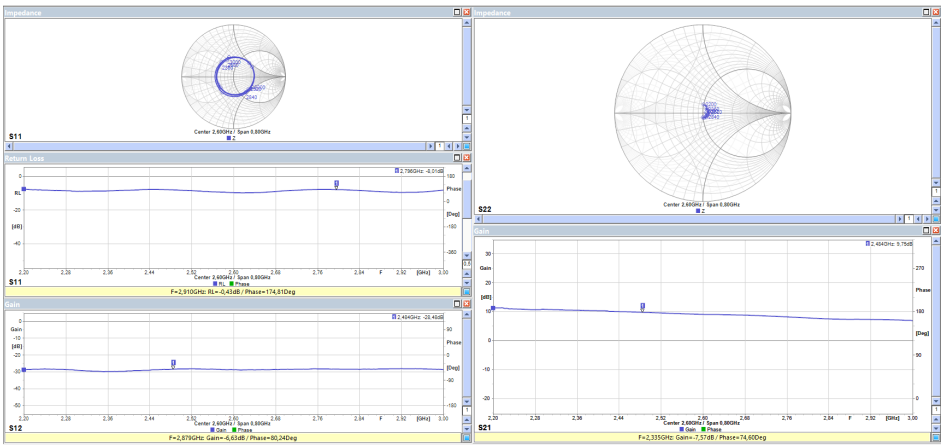


Figura 5.2: Caracterización del amplificador lineal con un voltaje de alimentación de 7.5V.(Medidas realizadas con el equipo VNA)

4. La salida del amplificador lineal se conecta al sensor de medición de potencia *NI USB-5681* (sección 4 de la Figura 5.1). Este sensor es el encargado de capturar la potencia de salida del amplificador lineal y enviarlo hacia un segundo ordenador.
5. Mediante el segundo ordenador en el cual se ejecuta el *software NI-568x Soft Front Panel* propio del sensor (sección 5 de la Figura 5.1) se muestra de forma digital el valor de la potencia de la señal leída a la salida del amplificador lineal.

5.1.2. Elección del HackRF ONE

Al momento de poner en funcionamiento el escenario implementado en la Figura 5.1 con los tres dispositivos HackRF, se obtiene la potencia de transmisión de los tres dispositivos HackRF al variar la frecuencia de transmisión. Estos resultados se presentan en la Figura 5.3.

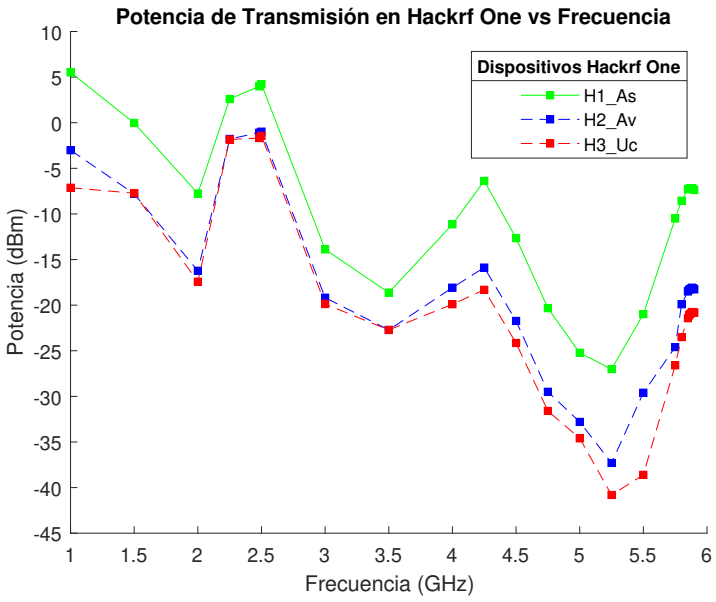


Figura 5.3: Potencia de transmisión de los dispositivos HackRF ONE en un rango de frecuencia de 1 a 6 GHz

En la Figura 5.3 se observa que, los dispositivos H2_Av y H3_Uc presentan un comportamiento similar salvo una pequeña variación de potencia en frecuencias de 1 GHz y en frecuencias entre 5 y 5.5 GHz. Mientras que, el HackRF etiquetado como H1_As presenta un mejor rendimiento en transmisión ya que, en frecuencias de interés como en 5.890 GHz (frecuencia a la que está establecido el estándar IEEE 802.11p) tiene una ganancia por encima de los -10 dBm. Por esta razón el HackRF ONE H1_As será el utilizado para la parte de transmisión mientras que el HackRF ONE H2_AV será el que se use para recepción.

5.1.3. Elección de la frecuencia de transmisión

El objetivo es identificar las frecuencias a las que el HackRF ONE elegido (H1_As) trasmite a mayor potencia. Además, verificar si el dispositivo es capaz de procesar la señal OFDM usando las frecuencias de transmisión identificadas.

En el programa realizado en GNU Radio (Capítulo 4) se puede elegir diferentes frecuencias de transmisión. Las frecuencias de interés son: 2.5 GHz y 5.8 GHz debido a que, en el programa de transmisión y recepción hay canales que están establecidos en frecuencias cercanas a las mencionadas anteriormente. Cercana a la frecuencia de 2.5 GHZ se encuentra el canal 14 (frecuencia de 2.484 GHz) y cercana a 5.8 GHz está el canal 178 (frecuencia de 5.89 GHZ).

Se procede a medir la potencia de transmisión del HackRF con y sin el uso del amplificador lineal. Se sigue el mismo procedimiento del escenario implementado en la Figura 5.1. En la Figura 5.4 se muestra los resultados.

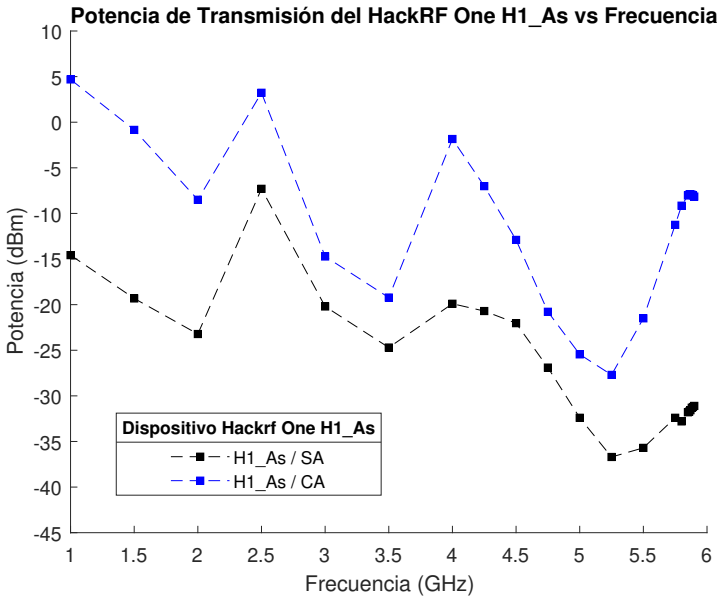


Figura 5.4: Potencia de transmisión del HackRF H1_As sin el uso del amplificador lineal (H1_As/SA) y con el uso del amplificador lineal (H1_As/CA)

Las líneas en color negro (sin amplificador lineal) muestran que, en la frecuencia 5.890 GHz (frecuencia de operación del estándar IEEE 802.11p) se alcanza una potencia de transmisión por debajo de los -30 dBm. Mientras que en la frecuencia de 2.500 GHz el dispositivo transmite a una potencia superior a los -10 dBm. Por otro lado, las líneas en color azul (con amplificador lineal) muestran que la potencia de la portadora ha aumentado, obteniendo potencias de alrededor de -5 dBm en la frecuencia de 5.890 GHz y por encima de los 0 dBm en la frecuencia de 2.500 GHz.

Tomando en consideración el comportamiento del HackRF en la frecuencia de 2.500 GHz y 5.890 GHz, se hace uso del programa de transmisión y recepción desarrollados en el Capítulo 4. Estos programas serán ejecutados en el HackRF `H1_As` (Transmisor) y en `H2_Av` (Receptor).

El objetivo es capturar los paquetes enviados desde el transmisor a una frecuencia de 5.890 GHz y 2.500 GHz con una separación entre el transmisor y receptor de 1 metro. Con la frecuencia 5.890 GHz (canal 178) y una distancia de separación de 1 metro no se logra capturar paquetes. Por lo que se da paso a la amplificación de la señal mediante el amplificador lineal. De esta manera se consigue la captura de paquetes, manteniendo la frecuencia y la distancia de separación.

Cuando se cambia de frecuencia de transmisión a 2.484 GHz (canal 14), se logra la captura de paquetes, manteniendo la distancia de 1 metro, sin la necesidad de usar el amplificador lineal. De esta manera se identifica que el HackRF tiene un mejor rendimiento en transmisión a una frecuencia de 2.484 GHz (canal 14).

Por los resultados obtenidos anteriormente se procede al cambio de frecuencia de transmisión de 5.890 GHz a 2.484 GHz, para las pruebas de campo dentro del escenario ideal y real (Capitulo 6). En la segunda frecuencia es en donde se podrá obtener mayores distancia de cobertura por parte de la RSU y es la frecuencia en donde el dispositivo trabaja de mejor manera.

5.2. Modulación y ancho de banda en transmisión

Una vez definida la frecuencia de transmisión, el siguiente paso es elegir el ancho de banda y modulación adecuado en donde el estándar funcione de manera correcta. Dado que el estándar IEEE 802.11p tiene establecido que, el ancho de banda a utilizar puede llegar a ser 10 MHz, el programa de transmisión tiene la disposición de utilizar 2, 5 y 10 MHz. Dentro del mismo programa se han establecido varias modulaciones las mismas que fueron mencionadas en el Capítulo 4.

Para obtener la modulación y ancho de banda óptimos, se implementó el escenario de la Figura 5.5. El escenario está formado por: un analizador de espectros, el amplificador lineal alimentado a 7.5V, el dispositivo HackRF ONE `H1_As` y el ordenador en donde se ejecuta el programa de transmisión.

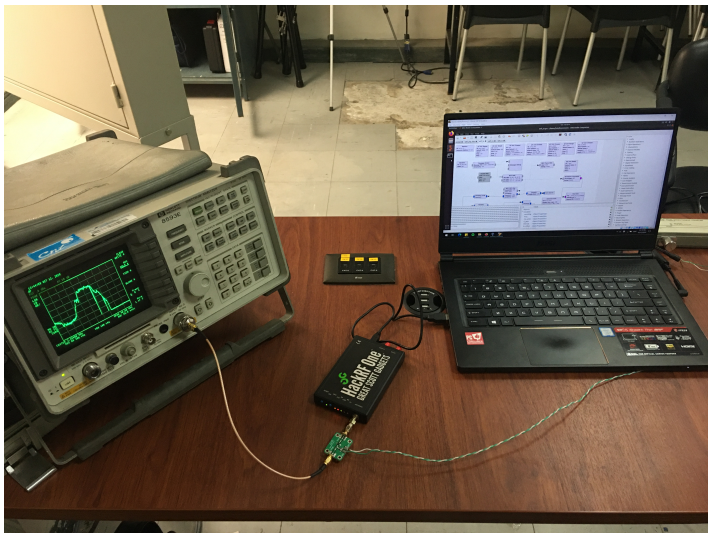


Figura 5.5: Escenario implementado para la selección de modulación y ancho de banda en transmisión

5.2.1. Selección de la modulación

Se ejecuta el programa de transmisión manteniendo fijo el ancho de banda a 2 MHz y variando el tipo de modulación. Al usar una modulación BPSK 1/2 se observa que el espectro de la señal OFDM se transmite sin dificultad la misma que es capturada por el analizador de espectros.

Cuando se usa una modulación BPSK 3/4 el espectro OFDM no se recibe de forma completa, por lo que es necesario activar la opción *Max Hold* en el analizador de espectros para que después de varias transmisiones se muestre el espectro de forma completa. Este problema en transmisión también se ve reflejada en otros tipos de modulaciones que se encuentran dentro de la lista de opciones a escoger como pueden ser QPSK o QAM.

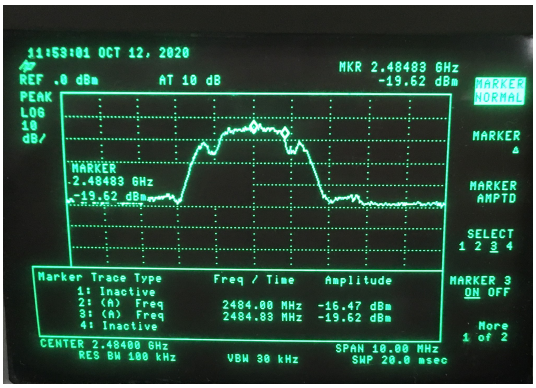
Se elige BPSK 1/2 como la modulación a utilizar para realizar la transmisión debido a que el espectro de la señal OFDM se transmite de forma completa, sin la necesidad de usar *Max Hold* en el analizador de espectro.

Hay que mencionar que en un experimento realizado y que se lo explicará en el Capítulo 6, se usa la modulación BPSK 1/2 en transmisión. Con esta modulación se logra conseguir un mayor número de paquetes recibidos por parte de la OBU, lo que no sucede cuando se usa otro tipo de modulación. La codificación en transmisión, la decodificación y detección del *frame* en recepción son tareas computacionales complejas que afectan al bloque Complex Programmable Logic Device (CPLD) y al procesador del HackRF ONE.

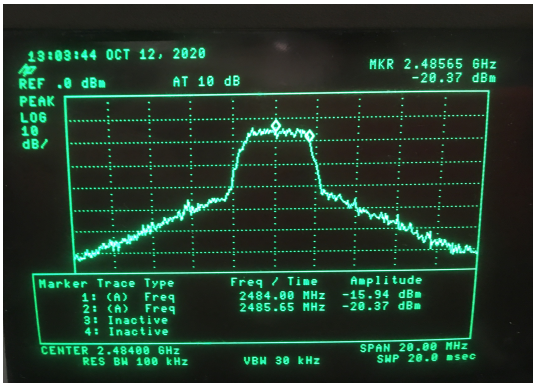
5.2.2. Elección de ancho de banda

Seleccionada la modulación, para elegir el ancho de banda ideal y verificar cual resulta más eficiente, se recurre nuevamente al escenario de la Figura 5.5. Entre los anchos de banda a elegir están: 2 MHz, 5 MHz y 10 MHz. Además, se configura el *span* en el equipo de medición con distintos valores (10, 20 y 40 MHz) para visualizar la gráfica del espectro OFDM de mejor manera.

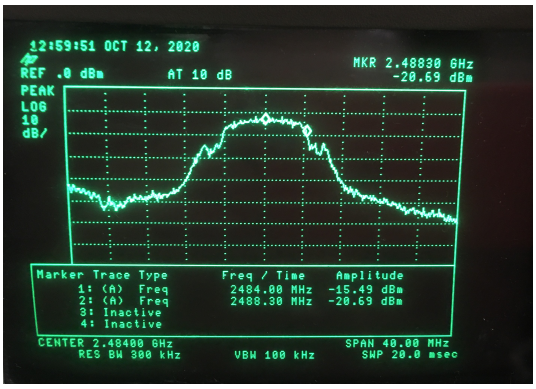
Cuando se transmite con un ancho de banda de 2 MHz, una frecuencia central de 2.484 GHz, y un *span* de 10 MHz se consigue la respuesta mostrada en la Figura 5.6a. De la misma manera, cuando se transmite con un ancho de banda de 5 MHz y un *span* de 20 MHz se consigue la respuesta mostrada en la Figura 5.6b. Y por último, cuando se transmite con un ancho de banda de 10 MHz y un *span* de 40 MHz se tiene la respuesta mostrada en la Figura 5.6c.



(a) Captura de la transmisión con un ancho de banda de 2 MHz



(b) Captura de la transmisión con un ancho de banda de 5 MHz



(c) Captura de la transmisión con un ancho de banda de 10 MHz

Figura 5.6: Anchos de banda en transmisión

En las tres gráficas de la Figura 5.6 se observa que los anchos de banda trasmitidos son cercanos a los que se espera de forma teórica, es decir, cumplen con el ancho de banda definido.

Al usar un ancho de banda de 5 y 10 MHz, el espectro de la señal OFDM no se transmite de forma completa. Este es un resultado similar a lo que sucede con una modulación superior a BPSK 1/2 (sección 5.2.1). Estas características son resultado de la limitación en procesamiento del dispositivo HackRF ONE.

La tarea computacional más compleja en transmisión es la codificación y un uso mayor de ancho de banda de la señal, lo que afecta al CPLD del HackRF ONE. El CPLD se usa a modo de interfaz o puente lógico entre

el procesador (ARM Cortex-M4/M0+) y los demás componentes del HackRF ONE.

Se usa nuevamente el sensor de potencia *NI USB-5681* con la intención de corroborar el problema expuesto anteriormente. En la Figura 5.7 se ven pulsos a una amplitud y tiempos de transmisión uniforme cuando el ancho de banda en transmisión es de 2 MHz, es decir, su espectro se transmite de forma completa.

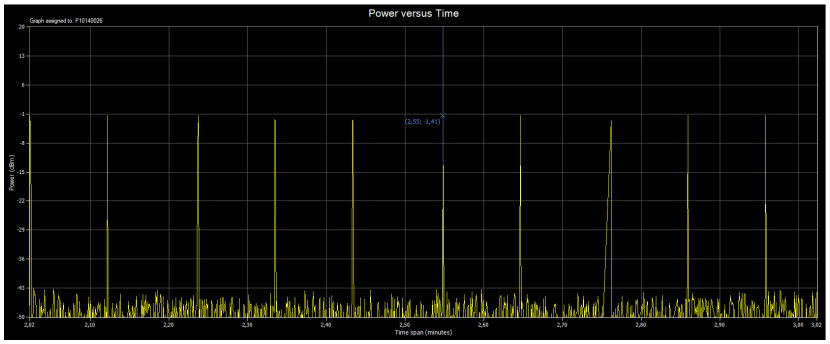
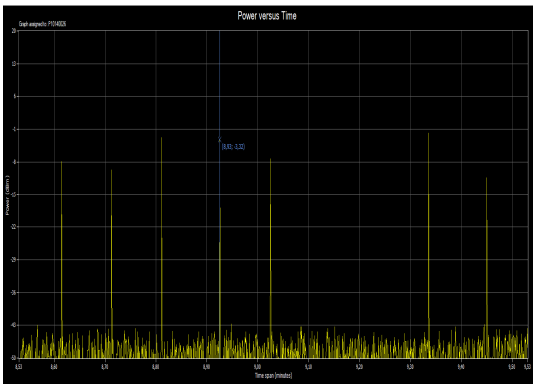
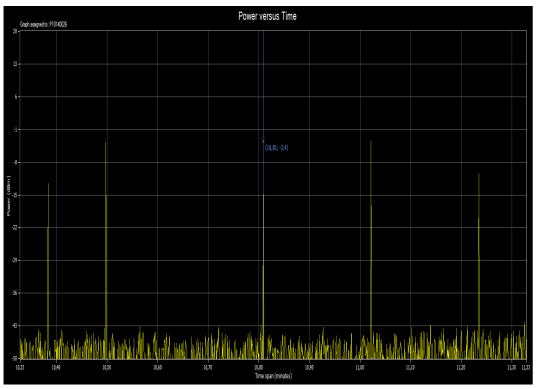


Figura 5.7: Potencia de transmisión cuando el ancho de banda es de 2 MHz capturada por el sensor de potencia NI USB-5681

Cuando se transmite con un ancho de banda a 5 MHz el tiempo entre una transmisión y la siguiente no se mantiene constante por lo que no se observa pulsos correspondiente a la señal de transmisión y al mismo tiempo esta señal pierde potencia, Figura 5.8a. Al enviar la señal con un ancho de banda de 10 MHz el tiempo entre transmisiones tampoco es constante al igual que la potencia de la señal, Figura 5.8b. Estas variaciones en potencia y tiempos entre transmisiones se ven reflejados en que el espectro OFDM no se transmite de forma completa.



(a) Potencia de transmisión cuando el ancho de banda es de 5 MHz



(b) Potencia de transmisión cuando el ancho de banda es de 10 MHz

Figura 5.8: Potencias de Transmisión capturas por el sensor NI USB-5681

El análisis anterior lleva a la conclusión de que el uso del ancho de banda a 2 MHz junto con la modulación BPSK es la mejor configuración soportada por el HackRF ONE, debido a que este procesa de manera correcta la señal OFDM. Como se muestra en la Figura 5.7 al utilizar los dos parámetros mencionados, el tiempo entre transmisiones y los niveles de potencia son uniformes. El dispositivo HackRF ONE tiene un procesamiento limitado y no es posible usar un esquema de modulación superior.

Las transmisiones empleando anchos de banda a 5 MHz y 10 MHz (Figura 5.8) no son recomendados debido a que en ocasiones el HackRF ONE no alcanza a realizar el procesamiento necesario para empezar el envío de paquetes, a pesar que para todas estas transmisiones se ha utilizado modulación BPSK tasa de $1/2$.

Luego de haber definido los valores a utilizar para la transmisión, los cuales son: modulación BPSK tasa 1/2 y ancho en de banda 2MHz, los parámetros que pueden variar serán *if gain* el cual actúa sobre el amplificador interno del HackRF y el voltaje de alimentación V_{in} del amplificador lineal usado. En la Tabla 5.1 se presenta los valores que pueden tomar estos parámetros junto con la potencia en transmisión obtenida. Hay que mencionar que al colocar valores menores a 36 en el parámetro *if gain* no se observa un aumento en potencia de transmisión. Los parámetros 4 y 5 son los que ofrecen una potencia de transmisión más elevada que los otros tres, por lo que serán potencias a usar para los experimentos posteriores y con los cuales se pretende conseguir una mayor cobertura por parte de la RSU.

Tabla 5.1: Parámetros a tomar en consideración para establecer la potencia de transmisión

Parámetros	If Gain	Vin (V)	Modulación	Ancho de Banda	Potencia en TX
1	36	0	BPSK 1/2	2 MHz	-32.7 dBm
2	50	0	BPSK 1/2	2 MHz	-12.36 dBm
3	36	5	BPSK 1/2	2 MHz	-23.67 dBm
4	50	5	BPSK 1/2	2 MHz	-1.9 dBm
5	50	7.5	BPSK 1/2	2 MHz	4.11 dBm

5.3. HackRF ONE en recepción

Para caracterizar el dispositivo HackRF ONE en recepción se implementó un nuevo escenario que se muestra en la Figura 5.9. En el cual se ha enumerado las diferentes etapas: envío de la señal OFDM, amplificación de la señal enviada por parte del amplificador lineal, atenuación y medición de la potencia y paquetes recibidos. No se considera un circuito de acoplamiento de impedancias debido a que los equipos manejan impedancias iguales a 50 *Ohm*. El proceso de caracterización se explica a continuación.

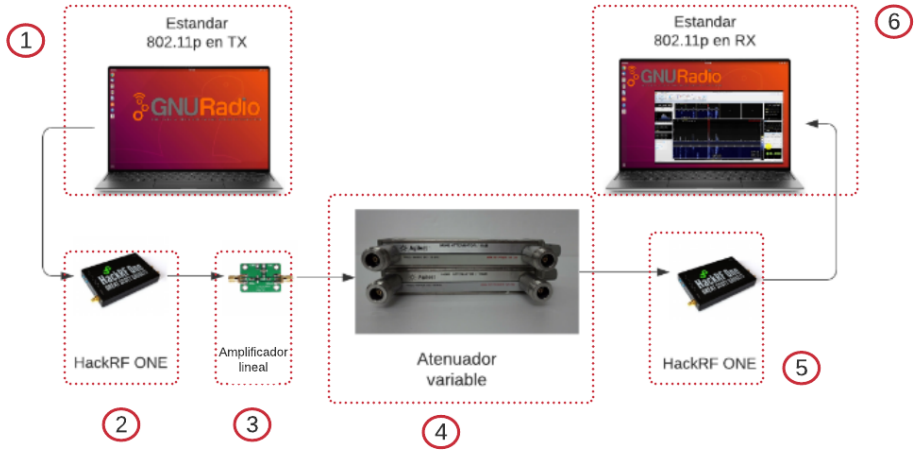


Figura 5.9: Escenario implementado para la caracterización del HackRF ONE en recepción

1. En la sección 1 de la Figura 5.9, está un ordenador en el cual se ejecuta el programa de transmisión, para el envío de paquetes.
2. A continuación se encuentra el HackRF ONE `H1_As` (sección 2 de la Figura 5.9) que procesa y envía la señal.
3. El siguiente elemento es el amplificador lineal alimentado con un voltaje de 7.5V (sección 3 de la Figura 5.9). A la salida del amplificador lineal se obtiene una potencia de transmisión de 4.11 dBm.
4. La señal amplificada desde el amplificador lineal es constante e ingresa a un atenuador variable (sección 4 de la Figura 5.9). Este dispositivo atenuará la señal de entrada de acuerdo a sus valores. El efecto que se busca es simular pérdidas, por lo que al aumentar el valor del atenuador, la potencia a su salida será menor.
5. A la salida del atenuador variable se encuentra el HackRF ONE etiquetado como `H1_Av` (sección 5 de la Figura 5.9).
6. El HackRF ONE `H1_Av` se conecta a un segundo ordenador en el cual se captura la potencia de la señal

recibida mediante el software *SDR Console* (sección 6 en la Figura 5.9. Además, se utiliza el programa para recepción del estándar IEEE 802.11p, con el cual se capturará paquetes. La potencia irá variando debido al atenuador, haciendo analogía a un comportamiento en un escenario real, en el cual se espera que al incrementar la distancia entre el transmisor y receptor, la potencia de la señal recibida vaya disminuyendo al igual que el porcentaje de paquetes recibidos por parte del dispositivo.

Los resultados de la potencia en recepción (dB) y paquetes recibidos (%) en función de la atenuación se muestran en la Figura 5.10.

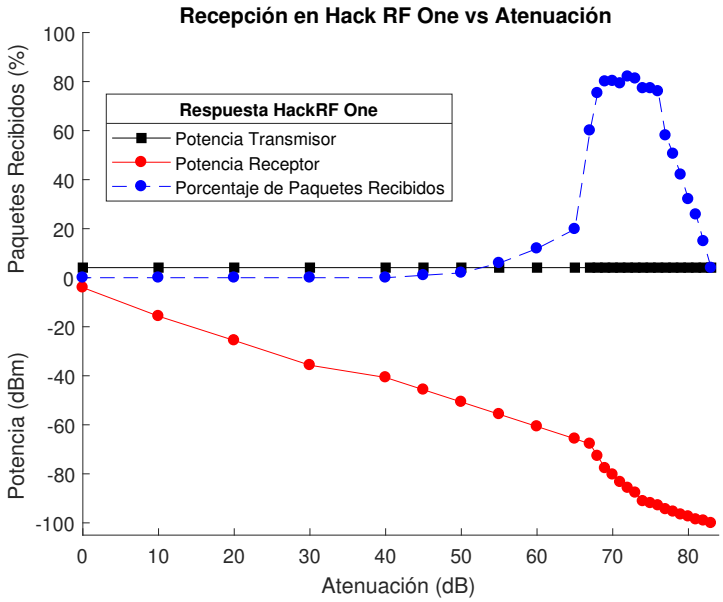


Figura 5.10: Respuesta en recepción del HackRF ONE en potencia y paquetes recibidos en función de la atenuación

Analizando las gráficas obtenidas (Figura 5.10) se observa que, cuando se alcanza una atenuación de 40 dB la potencia en recepción disminuye y el porcentaje de paquetes recibidos es del 0 %. Lo que da a entender que el HackRF ONE se satura a potencias mayores a los -40 dBm.

Con una atenuación de 45 dB la potencia en recepción también decrece pero el porcentaje de paquetes recibidos es de alrededor del 1 %, es decir el HackRF ya no se encuentra en saturación.

Estos resultados muestran que el HackRF ONE `H1_Av` se satura a una potencia mayor de -40 dBm. Desde -45 dBm a -60 dBm se empieza a recibir paquetes aunque no de una forma eficiente por lo que su porcentaje en paquetes recibidos es menor al 20 %. Cuando la potencia recibida está en el rango de -60 dBm hasta los -95 dBm el porcentaje de paquetes recibidos va en aumento alcanzando un porcentaje del 80 %. Si la potencia en recepción es menor de -95 dBm el porcentaje de paquetes recibidos empieza a decrecer.

5.4. Conclusiones

La caracterización del HackRF ONE en transmisión consiste en medir su potencia de salida. Esta permitió seleccionar el dispositivo que mejor rendimiento posee entre los tres que se tiene a disposición. Se identificó en que frecuencias de transmisión se alcanzan las mayores potencias y el mejor rendimiento.

Con la caracterización del HackRF ONE junto con el amplificador lineal, se observa el aumento de potencia de transmisión del dispositivo y al mismo tiempo se identifica las frecuencias en la que el HackRF y el amplificador lineal tienen su mejor rendimiento. El HackRF usado en recepción (`H1_Av`), a una frecuencia de transmisión de 2.484 GHz, logra la captura de paquetes a distancias superiores a 1 metro de separación entre la RSU y la OBU, resultado que no se obtiene al usar la frecuencia de 5.890 GHz. Cuando se usa esta última frecuencia no se sobrepasa el metro de distancia en la recepción de paquetes. Este resultado se obtuvo al aumentar de manera progresiva la distancia de separación entre la RSU y la OBU. Se empieza por los 10 centímetros con incrementos de 10 y se obtiene un metro como el límite para la recepción de paquetes al usar la frecuencia de 5.9 GHz. Como conclusión de este experimento, la frecuencia de 2.484 GHz es en donde el HackRF ONE tiene el mejor rendimiento en transmisión y será la frecuencia usada para pruebas.

En transmisión se han definido los parámetros de BPSK 1/2 para modulación y 2MHz para el ancho de banda. Con los parámetros antes mencionados, el espectro de la señal OFDM en transmisión se forma correctamente y su potencia se mantiene constante. Lo que no sucede al utilizar otro tipo de modulación o un ancho de banda diferente, debido a que el HackRF ONE no puede realizar el procesamiento necesario. Este es un problema que se refleja en el espectro y en la potencia de transmisión, como se lo mencionó en la sección 5.2.1.

Con la caracterización del HackRF ONE en recepción y mediante el escenario de la Figura 5.9 (escenario implementado para la caracterización del HackRF ONE en recepción) se obtiene la gráfica de la Figura 5.10 (Respuesta del HackRF ONE en recepción). Con estos resultados se observa que, cuando el HackRF ONE recibe a potencias superiores a los -40 dBm, este entra en un estado de saturación por lo que no habrá paquetes recibidos. Existe un rango de potencia entre los -65 dBm y los -95 dBm en donde el comportamiento del HackRF es eficiente demostrando que puede recibir un porcentaje de paquetes de alrededor del 80 %. No se obtiene un 100 % de paquetes recibidos debido a que el HackRF ONE recibe los paquetes en una cola y esta no despacha los paquetes a la misma velocidad que los recibe, es por eso que los últimos paquetes en llegar no son procesados.

Capítulo 6

Pruebas realizadas y resultados obtenidos

Teniendo en cuenta todos los factores analizados en el Capítulo 5, sobre el comportamiento del HackRF ONE a distintas frecuencias, se procede a realizar las pruebas estáticas y en movimiento. Se van a establecer diferentes pruebas, primero se realizan pruebas en un escenario ideal de forma estática. Luego, se realizan pruebas en un escenario real, en calles con circulación de vehículos y peatones. En los dos escenarios, se establece la conexión de la RSU que estará montada en un semáforo y de la OBU que se colocará dentro de un vehículo. La cobertura que se alcanza en los diferentes escenarios se la define como la distancia máxima a la que se reciben paquetes.

6.1. Pruebas y resultados en escenario ideal

Como se ha mencionado anteriormente, las pruebas están realizadas en diferentes escenarios. El escenario ideal que se utiliza para realizar las pruebas, es un terreno con un campo abierto, Figura 6.1, resaltado por la línea en color rojo.

Figura 6.1: Escenario ideal para pruebas del Estándar IEEE 802.11p

El escenario ideal es un área libre de obstáculos. La señal no tiene interferencias por objetos que se encuentran

entre el transmisor y el receptor. Las pruebas se realizan con el transmisor y el receptor colocados a una altura de 1 m y 0.6 m respectivamente. En la Figura 6.2 se muestra como está ubicado el receptor y el transmisor.

Figura 6.2: Pruebas de transmisión y recepción en escenario estático ideal

6.1.1. Potencia recibida por parte del HackRF ONE

Se empieza por medir la potencia en recepción del HackRF ONE usando el *software SDR Console*. Para esto el transmisor envía la señal OFDM a cuatro diferentes potencias. Los valores de potencia en el receptor con sus intervalos de confianza se muestran en la Figura 6.3. Cabe mencionar que el amplificador lineal se usó en transmisión, en recepción no se conectó un amplificador lineal. Los resultados muestran un comportamiento en donde la potencia de la señal OFDM recibida disminuye a medida que la distancia aumenta. Dentro de este escenario se obtiene mediciones de potencias superiores a los -85 dBm a 90 metros de distancia, cuando la potencia en transmisión es de 4.11 dBm. En función de estos resultados se alcanzarán distancias de cobertura de alrededor de 90 metros al pasar a escenarios reales.

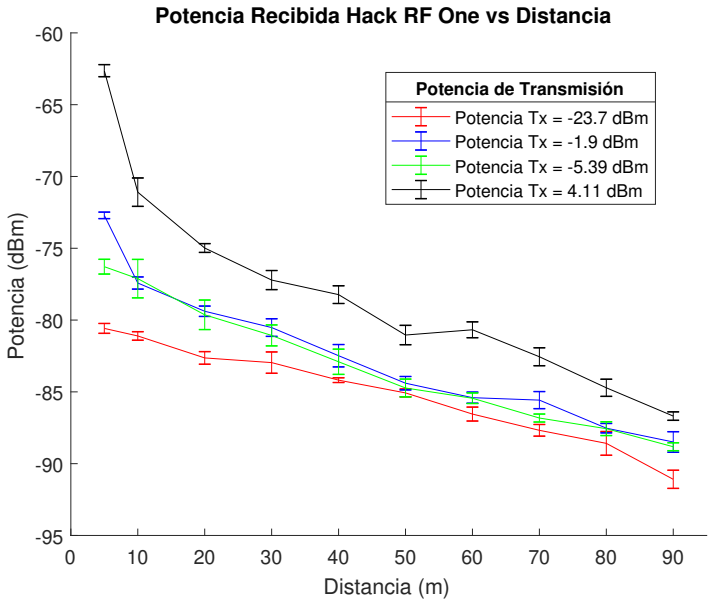


Figura 6.3: Potencia medida en recepción al variar distancia dentro del escenario estático ideal

6.1.2. Comparación con esquemas de modulación

En esta sección se analiza el comportamiento en recepción de paquetes recibidos y retardo (*delay*). El comportamiento se analiza con una señal OFDM y las modulaciones BPSK y QPSK con tasas de transmisión de 1/2 y 3/4. La potencia de transmisión se mantiene constante con un valor de -23.67 dBm. Con esta potencia se logra una cobertura de 60 metros. Se intentó capturar paquetes a una distancia superior pero no se obtuvo resultados al usar la potencia mencionada (-23.67 dBm).

6.1.2.1. Análisis del porcentaje de paquetes recibidos

Para calcular el porcentaje de paquetes recibidos se usan los archivos `pcap` en la RSU que contiene el número de paquetes transmitidos. De la misma manera en el archivo `pcap` de la OBU se encuentra el número de paquetes recibidos. El porcentaje de paquetes recibidos es igual al número de paquetes recibidos por la OBU sobre el número de paquetes enviados por la RSU.

En la Figura 6.4 se muestra el porcentaje de paquetes recibidos al variar su esquema de modulación y su tasa de transmisión. Con la modulación BPSK 1/2 y a una distancia de 5 metros, se obtiene un porcentaje de paquetes recibidos superior al 80 %, el mismo que desciende a medida que la distancia aumenta. A una distancia de 60 metros el porcentaje de paquetes recibidos es menor al 5 %. Con BPSK 1/2, al ser una modulación más robusta, se alcanza los 60 metros, lo que no sucede con las demás modulaciones.

Con la modulación BPSK 3/4 se alcanza una cobertura de 30 metros. A los 10 metros se obtiene un porcentaje de paquetes recibidos de alrededor del 80 %. A una distancia mayor a 10 metros el porcentaje de paquetes disminuye considerablemente, hasta alcanzar un 10 % en paquetes recibidos a los 30 metros. Usando QPSK 1/2 se alcanza una cobertura de 20 metros, se obtiene un porcentaje de paquetes recibidos por debajo del 60 %. Con QPSK 3/4 se logra una cobertura de 10 metros, obteniendo un porcentaje de paquetes recibidos similar a la anterior modulación.

Al aumentar el esquema de modulación el porcentaje de paquetes recibidos disminuye al igual que la distancia

de cobertura. Esto es debido a que, la codificación que se realiza en el transmisor es una tarea computacional compleja. En la recepción la detección del *frame* junto con la decodificación son las tareas que más recursos computacionales demandan hacia el procesador y los componentes del HackRF ONE. Además no existen trabajos previos en donde se use este dispositivo para transmitir un estándar completo y ver su comportamiento en transmisión y recepción.

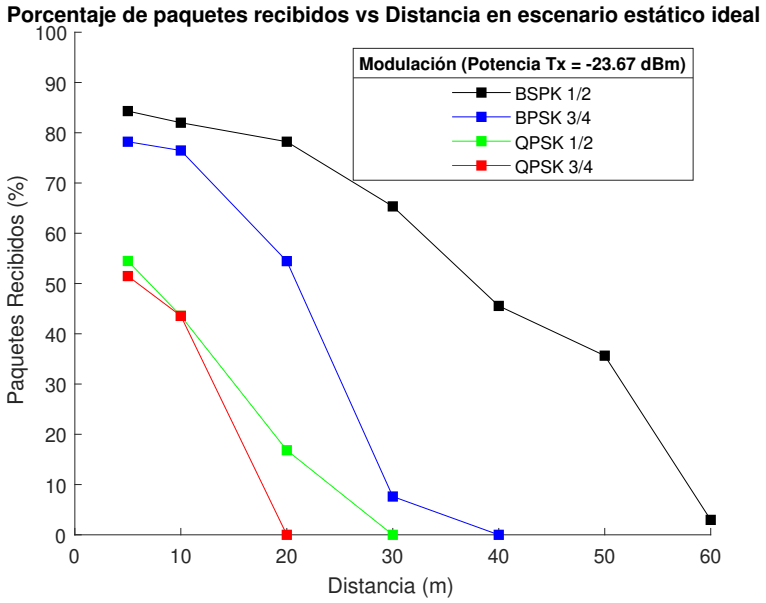


Figura 6.4: Porcentaje de paquetes recibidos al variar distancia y modulación en el escenario estático ideal

6.1.2.2. Análisis del retardo

Para encontrar el retardo en los paquetes recibidos, tanto para el escenario ideal y escenarios reales (estáticos y en movimiento), primero fueron sincronizados la RSU y la OBU. En el `pcap` de la RSU se tiene la hora, minutos y segundos en el que el paquete fue generado y se transmite. De igual manera en el `pcap` generado en la OBU, se cuenta con los datos relacionados al tiempo en que el paquete fue capturado. Otro campo importante dentro de los `pcap` es el número de secuencia (*sequence number*). El número de secuencia permite identificar a cada paquete, tanto en transmisión como en recepción.

Para encontrar el retardo entre los paquetes, se calcula la diferencia de tiempo entre la hora en el que el paquete se registra en el `pcap` de la OBU, menos la hora en el que el paquete se registra en el `pcap` de la RSU. Por último se calcula el promedio entre todos los valores de tiempo obtenidos en una transmisión. En el retardo calculado se encontrará el tiempo en el que el paquete atraviesa el medio inalámbrico, más el tiempo en el que el paquete es procesado y colocado en el `pcap` por parte del HackRF. Los resultados se muestran en la Figura 6.5.

Como se muestra en la Figura 6.5 al utilizar una modulación BPSK 1/2, se obtiene un retardo por debajo de los 2 segundos. Mientras que para las demás modulaciones el retardo es superior a los 2 segundos. Este retardo se debe al procesamiento del dispositivo como se mencionó en la sección 6.1.2.1, la codificación y decodificación son las tareas más complejas a realizar. Se puede analizar hasta los 30 metros para BPSK 3/4, 20 metros para QPSK 1/2 y 10 metros para QPSK 3/4 debido a que, cuando se supera las distancias mencionadas no se cuenta con paquetes recibidos.

El retardo en BPSK 1/2 disminuye a medida que la distancia aumenta. A pesar de los resultados mostrados

en la Figura 6.4, se observa un menor porcentaje de paquetes recibidos a distancias mayores. Esto se evidencia a una distancia de 50 y 60 metros. El retardo es menor debido a que hay menos paquetes recibos, pero estos llegaron con retardos cercanos a 0.8 segundos.

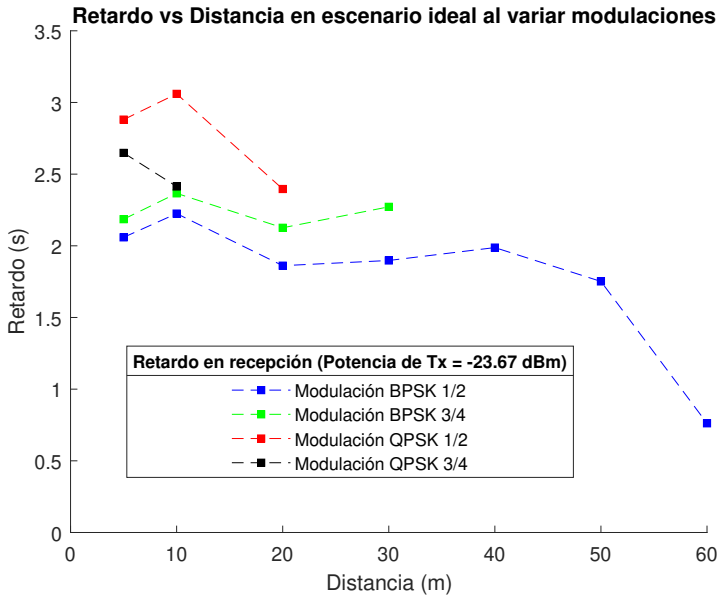


Figura 6.5: Retardo en función de la distancia y la modulación en el escenario estático ideal

6.1.3. Comparación con potencias de transmisión

Dentro del escenario ideal se analiza el comportamiento en recepción de los paquetes recibidos y el retardo. El análisis se realiza cuando se envía la señal OFDM. La señal se transmite a diferentes potencias, tomando en consideración tres valores de potencia -23.67 dBm, -1.9 dBm y 4.11 dBm. La manera en que se estableció estas potencias se las detalla en la Tabla 5.1. En este experimento se alcanza una cobertura de 80 metros a causa del aumento de potencia.

6.1.3.1. Análisis del porcentaje de paquetes recibidos

Los resultados de este experimento se muestran en la Figura 6.6. Cuando se transmite la señal OFDM a una potencia de -23.67 dBm se consigue un porcentaje de paquetes recibidos por encima del 80 % a distancias de 5 y 10 metros. A medida que la distancia aumenta, el número de paquetes recibidos disminuye siendo la distancia máxima de cobertura 60 metros (3 % de paquetes recibidos).

Al realizar la transmisión con -1.9 dBm se obtiene un 80 % de paquetes recibidos a una distancia de 5 metros. A los 10 metros un 70 % de paquetes recibidos. Desde los 10 metros en adelante este porcentaje disminuye hasta alcanzar un 10 % a una distancia de 80 metros.

Al realizar la tercera transmisión con una potencia de 4.11 dBm, se obtiene resultados diferentes a las dos transmisiones anteriores. Como se muestra en la Figura 6.6, a distancias de 5, 10 y 20 metros el porcentaje de paquetes recibidos se encuentra por debajo del 40 %. Este resultado contrasta con los datos obtenidos en las dos transmisiones anteriores. A una distancia de 30 metros el porcentaje de paquetes recibidos llega a valores por encima del 70 %, se tiene el pico más alto en los 80 metros con alrededor del 80 % de paquetes recibidos. Desde los 60 metros tiende a decrecer.

Estos resultados muestran que, el HackRF ONE entra en un estado de saturación cuando se transmite a una potencia elevada (4.11 dBm para este caso) y a distancias menores a los 30 metros. Lo cual provoca que su comportamiento en recepción no resulte efectivo. Por otro lado, cuando se llega a los 30 metros, el comportamiento en recepción muestra un mejor desempeño y se logra alcanzar los 80 metros de cobertura en el escenario ideal.

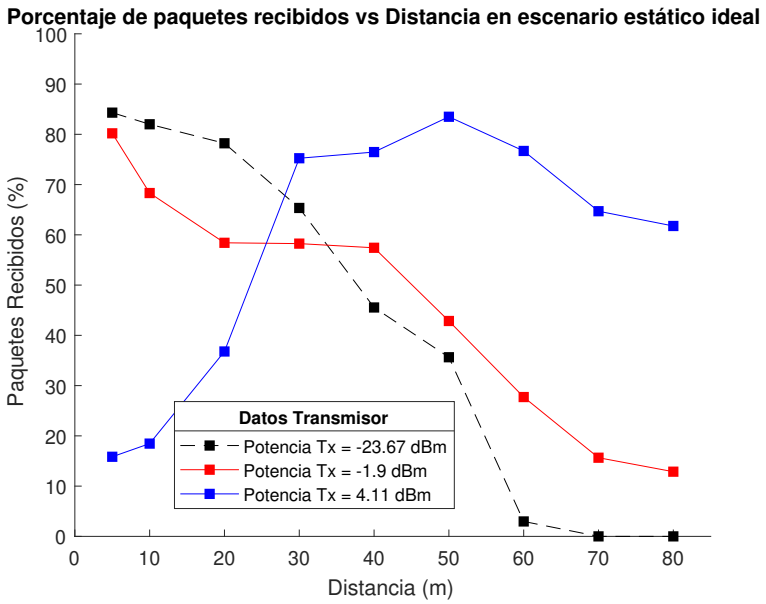


Figura 6.6: Porcentaje de paquetes recibidos al transmitir a diferentes potencias en el escenario ideal

6.1.3.2. Análisis del retardo

En la Figura 6.7 se muestran los resultados del retardo en función de la distancia, para las tres potencias propuestas. A una potencia de -23.67 dBm el retardo fluctúa entre los 2.5 y 3 segundos, con una distancia de cobertura de 60 metros.

A una potencia de -1.9 dBm, el retardo es de 2.5 segundos a una distancia de 5 metros, y desciende hasta llegar a un valor aproximado de 1.5 segundos a una distancia de 80 metros. Al transmitir a una potencia de 4.11 dBm, el retardo que se observa en la Figura 6.7 se encuentra por debajo de los dos anteriores, fluctuando entre 1 y 1.5 segundos dentro de los 80 metros de distancia.

Un resultado interesante en este punto es que mientras mayor es el porcentaje de paquetes perdidos, menor es el retardo. Este comportamiento se debe al número de paquetes que se reciben y al tiempo que les toma en llegar desde la RSU hasta la OBU, como se explicó anteriormente este valor se lo encuentra al promediar el retardo de varios paquetes dentro de una transmisión.

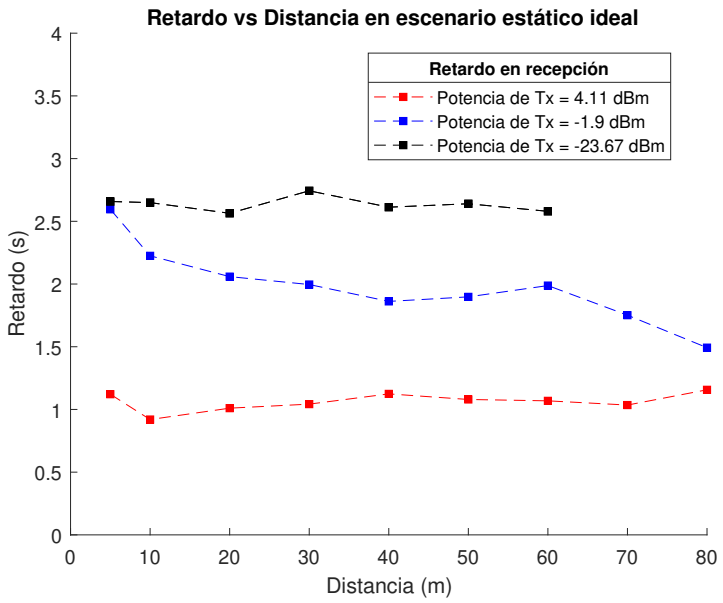


Figura 6.7: Retardo al transmitir a diferentes potencias en el escenario ideal

6.2. Pruebas y resultados en escenarios reales

Dentro de las pruebas a realizar, se cuenta con dos escenarios reales, en los cuales la RSU estará montada en un poste, a una altura de 3.15 metros medidos desde el suelo y la OBU se encuentra en un vehículo a una altura de 1.1 metros. Los equipos montados en los escenarios se muestran en la Figura 6.8.

En la parte izquierda de la Figura 6.8 se muestra el primer escenario. La calle no presenta curvas y es una carretera plana, a la cual se la denomina como `Escenario 1`. En la parte derecha de la Figura 6.8 se muestra el segundo escenario. Este presenta una curva no tan pronunciada con una pendiente de subida, al cual se le denomina `Escenario 2`. Igualmente en las Secciones 6.2.1 y 6.2.2 se muestra una imagen satelital obtenida de *Google Earth* correspondiente a los dos escenarios. Estos dos escenarios son los utilizados para pruebas estáticas y en movimiento.

Figura 6.8: RSU y OBU en escenarios para pruebas

6.2.1. Pruebas y resultados en Escenario 1 estático

El `Escenario 1` estático se encuentra formado por una carretera recta de dos carriles de ida y dos de vuelta. Para estas pruebas se usa el carril de ida, además la vía no presenta variaciones de altura. En el trayecto analizado circulan buses y automóviles de forma concurrida, el escenario se muestra en la Figura 6.9.

Figura 6.9: Escenario 1 estático

6.2.1.1. Comparación con potencias de transmisión

En el `Escenario 1` estático se analiza el comportamiento en recepción de los paquetes recibidos y el retardo, cuando se envía la señal OFDM a diferentes potencias de transmisión. En este escenario se toma en consideración dos valores de potencia -1.9 dBm y 4.11 dBm. Con estas dos potencias, se tiene una cobertura de 80 metros en el escenario estático ideal. Con la potencia de -23.67 dBm (la cual no se consideró) solo se alcanzó una cobertura de 60 metros. La manera en que se estableció estas potencias se las detallada en la Tabla 5.1.

6.2.1.2. Análisis del porcentaje de paquetes recibidos y retardo

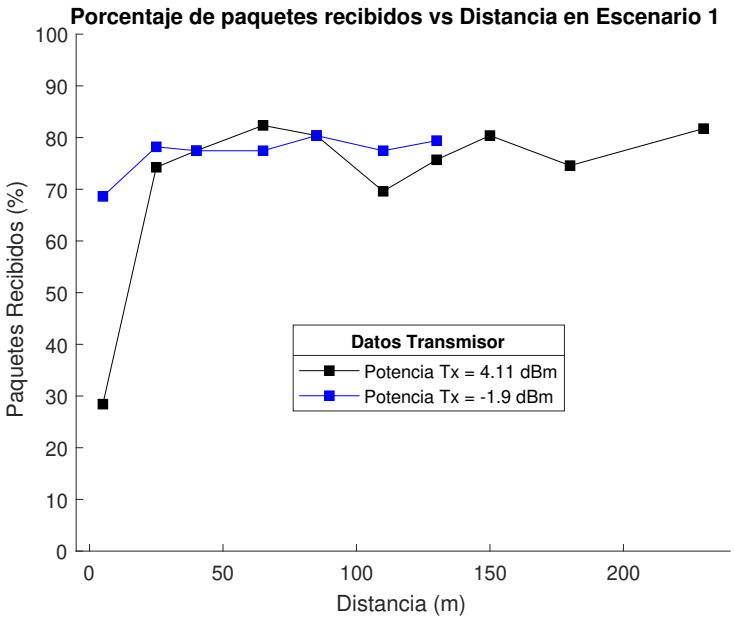
Los resultados obtenidos de porcentaje de paquetes recibidos se muestran en la Figura 6.10a. Dentro del `Escenario 1` estático se procede al envío de la señal a una potencia de -1.9 dBm. A los 5 metros de distancia se obtiene un porcentaje de paquetes recibidos cercano al 70 %. La siguiente distancia a la que se realiza la captura es de 25 metros, se logra un porcentaje de paquetes recibidos de alrededor del 80 %. Este porcentaje se mantiene hasta los 130 metros que es la máxima distancia que se logra cubrir al transmitir con esta potencia.

Al aumentar la potencia de transmisión a 4.11 dBm el resultado en paquetes recibidos a los 5 metros es menor al 30 %. Ocurre lo mismo que en el escenario ideal, el HackRF ONE se satura cuando se encuentra a distancias pequeñas de separación, en este caso 5 metros. Al realizar las siguientes capturas e ir aumentando la distancia de separación y manteniendo la potencia de 4.11 dBm, se logra cubrir una mayor distancia. En este experimento se alcanza los 230 metros de cobertura. El porcentaje de paquetes recibidos se mantiene entre un 70 % y 80 %.

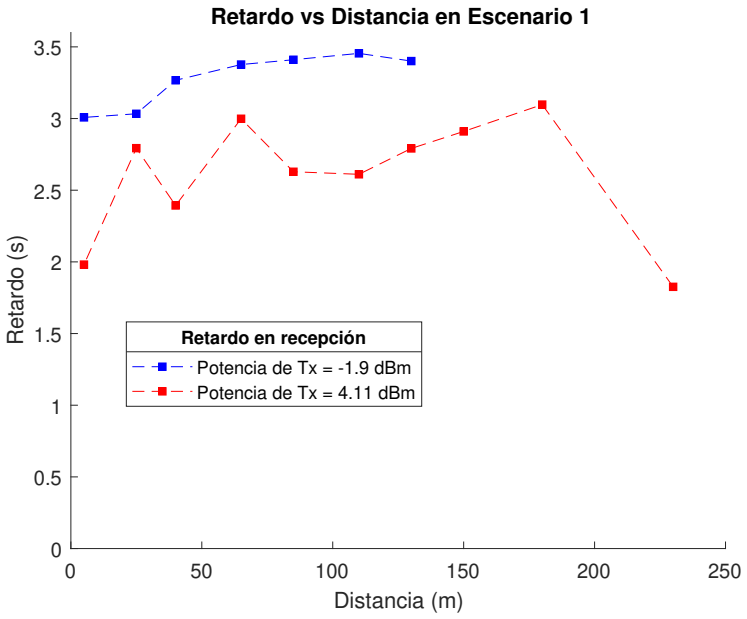
Estos resultados llevan a la conclusión de que, el HackRF se satura a distancias cortas entre la OBU y la RSU, cuando la potencia de transmisión es de 4.11 dBm. Pero se logra cubrir una mayor distancia entre la RSU y la OBU, en este caso 230 metros. Según especificaciones que se encuentran en la página oficial del HackRF ONE, es recomendable trabajar con potencias en recepción que no superen los -20 dBm, para proteger el amplificador de recepción. En esta recomendación, no se especifica que tipo de programa usa para obtener estas referencias.

En los experimentos realizados, la potencia a la que el HackRF ONE deja su estado de saturación es a potencias de recepción inferiores a los -45 dBm.

El retardo obtenido al variar potencia de transmisión y distancia se observa en la Figura 6.10b. Cuando la potencia en transmisión es de -1.9 dBm el retardo se encuentra alrededor de los 3 segundos a una distancia de 5 metros. Este retardo aumenta a medida que se incrementa la distancia. El retardo alcanza un pico máximo de 3.5 segundos a los 130 metros, distancia que se cubre usando esta potencia de transmisión. Para una potencia de transmisión de 4.11 dBm se observa que el retardo en recepción está por debajo de los 3 segundos. Con esta potencia se tiene un mejor rendimiento que con la potencia de transmisión de -1.9 dBm. Si bien el retardo a una potencia de 4.11 dBm está por debajo de los 3 segundos, este no se estabiliza en un valor determinado. Se resalta que, cuando la distancia entre transmisor y receptor es de 230 metros, se alcanza el menor valor de retardo (1.8 segundos). A esta distancia también se alcanzó un 80 % de porcentaje de paquetes recibidos (Figura 6.10a).



(a) Porcentaje de paquetes recibidos en función de la distancia y potencia de transmisión en el Escenario 1 estático



(b) Retardo en función de la distancia y potencia de transmisión en el Escenario 1 estático

Figura 6.10: Porcentaje de paquetes recibidos y retardo en el Escenario 1 estático

El `Escenario 1` estático presenta un flujo variado de vehículos y autobuses en la vía. Se evidencia en la Figura 6.10b a una distancia de 25, 65 y 180 metros picos en el valor del retardo, estos valores son debido a la presencia de autobuses en la vía. Los autobuses obstaculizan a la señal y por ende aumenta el retardo. Los valores de retardo a excepción del último (distancia de 230 metros) son ocasionados por el transito en la vía. El valor más bajo de retardo se obtiene a 230 metros, es la posición en la que la OBU y la RSU tienen la mejor linea de vista, no hay presencia de obstáculos y no hay paso de vehículos ni autobuses.

6.2.2. Pruebas y resultados en Escenario 2 estático

El `Escenario 2` estático contiene un carril de ida y uno de vuelta, además cuenta con una pendiente, y una curva no tan pronunciada. La circulación de buses y automóviles no es de forma concurrida. El `Escenario 2` estático se muestra en la Figura 6.9. En este escenario se analizan la transmisión de la señal OFDM a diferentes modulaciones.

Figura 6.11: Escenario 2 estático

6.2.2.1. Comparación con esquemas de modulación

En el `Escenario 2` estático las pruebas se las realiza con una distancia máxima de 70 metros, separadas en 4, 40, 60 y 70 metros debido a la geografía del terreno. La potencia en transmisión se mantiene constante en -1.9 dBm y se varía el esquema de modulación y la tasa de transmisión.

6.2.2.2. Análisis del porcentaje de paquetes recibidos

En la Figura 6.12 se muestra el porcentaje de paquetes recibidos al variar el esquema de modulación, la distancia y manteniendo la potencia de transmisión en -1.9 dBm. Con una modulación BPSK 1/2 se obtiene un porcentaje de paquetes recibidos de alrededor de 80 % en las 3 primeras distancias, y disminuye a un 55 % a 70 metros. La modulación BPSK 1/2 es la única modulación con la que se recibe paquetes a una distancia de 70 metros. Esta modulación permite alcanzar la mayor cobertura en el `Escenario 2`.

Con la modulación BPSK 3/4 se cubre una distancia de 60 metros, con un porcentaje de paquetes recibidos que disminuye a medida que se llega a esta distancia. Con la modulación QPSK 1/2 se tiene una cobertura de 60 metros, se obtiene un porcentaje de paquetes del 50 % a 4 metros y desciende hasta llegar a 20 %. Finalmente, con la modulación QPSK 3/4 solo se logra una cobertura de 4 metros y alrededor del 5 % de porcentaje de paquetes recibidos. Tanto la codificación como la decodificación son tareas computacionales complejas para el procesador del dispositivo y su bloque CPLD. La demanda computacional aumenta a medida que se utiliza codificaciones superiores.

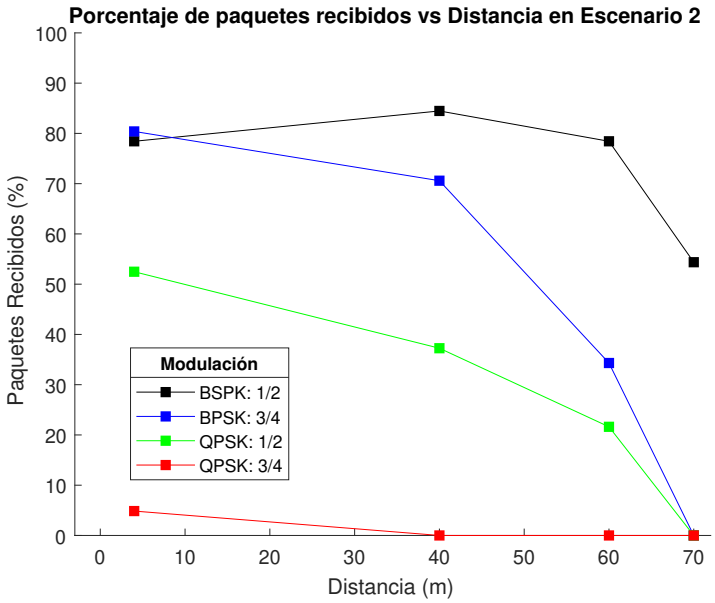


Figura 6.12: Porcentaje de paquetes recibidos en función de la distancia y de la modulación en el Escenario 2 estático

6.2.2.3. Análisis del retardo

La Figura 6.13 muestra los resultados del retardo en el `Escenario 2` estático. Al utilizar una modulación BPSK 1/2 se obtiene un retardo entre 0.5 y 1 segundo en los 70 metros de cobertura.

Con una modulación BPSK 3/4 el retardo incrementa su valor a 1.5 segundos. Utilizando QPSK 1/2 el retardo se encuentra por encima de los 2 segundos, aunque presenta una disminución a una distancia de 60 metros. Esta distancia es la máxima que se cubre con estas dos modulaciones. En cuanto al retardo, con una modulación QPSK 3/4 se tiene un valor por encima de los 4 segundos a una distancia de 4 metros. Esta distancia es la máxima que se cubre con esta modulación.

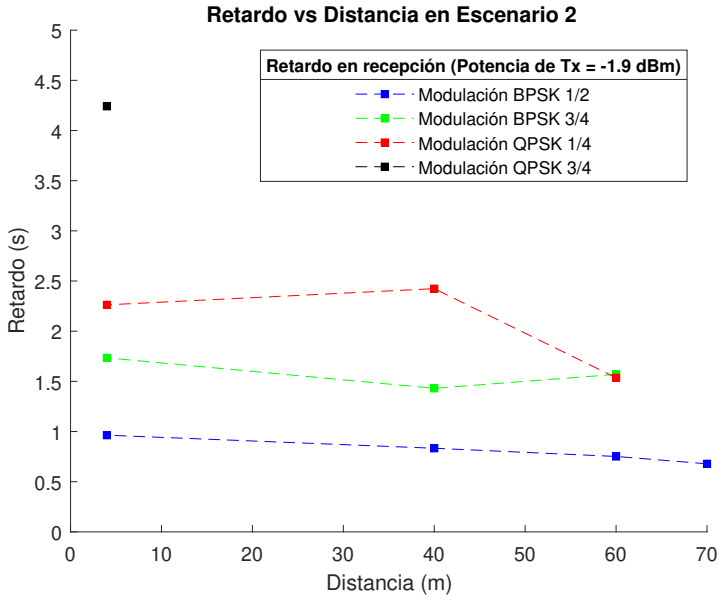


Figura 6.13: Retardo en función de la distancia y modulación en el Escenario 2 estático

6.2.3. Pruebas y resultados en escenarios móviles

Para estas pruebas se usaron los escenarios 1 y 2. Debido a la geografía del `Escenario 1` el vehículo alcanza velocidades de hasta 70 km/h. En este escenario se realiza la transmisión a dos potencias: -1.9 dBm y 4.11 dBm. En el `Escenario 2` el vehículo alcanza una velocidad máxima de 50 Km/h. En este escenario la transmisión se la realiza a una potencia de -1.9 dBm.

6.2.3.1. Análisis del porcentaje de paquetes recibidos

En la Figura 6.14 se muestra el porcentaje de paquetes recibidos en los escenarios móviles en función de la potencia de transmisión y velocidad del vehículo. Primero se analiza el `Escenario 1`. Se utiliza una potencia de transmisión de -1.9 dBm y el vehículo se desplaza a 20 y 30 Km/h. Con los datos mencionados, el porcentaje de paquetes recibidos es del 70 %. A medida que la velocidad aumenta desde los 40 Km/h hasta los 70 Km/h, el porcentaje de paquetes disminuye. Estos resultados son esperados debido a que el vehículo cada vez incrementa su velocidad y por lo tanto se aproxima de manera más rápida hacia la RSU.

Dentro del `Escenario 1` con una potencia de transmisión de 4.11 dBm, existe un 70 % de paquetes recibidos a velocidades de 20 y 30 Km/h. Este porcentaje decrece a medida que aumenta la velocidad. Cuando el vehículo se desplaza con las velocidades mencionadas, existe un incremento mínimo en el porcentaje de paquetes recibidos al enviar la señal a una potencia de 4.11 dBm que a una potencia de -1.9 dBm. Mientras que la tendencia a disminuir el porcentaje de paquetes recibidos conforme aumenta la velocidad se mantiene.

En el `Escenario 2` el vehículo se mueve a una velocidad mínima de 20 Km/h y una máxima de 50 Km/h. A una velocidad de 20 Km/h el porcentaje de paquetes recibidos está por encima del 50 %. La tendencia al igual que en el `Escenario 1` es de disminuir el porcentaje de paquetes recibidos a medida que aumenta la velocidad.

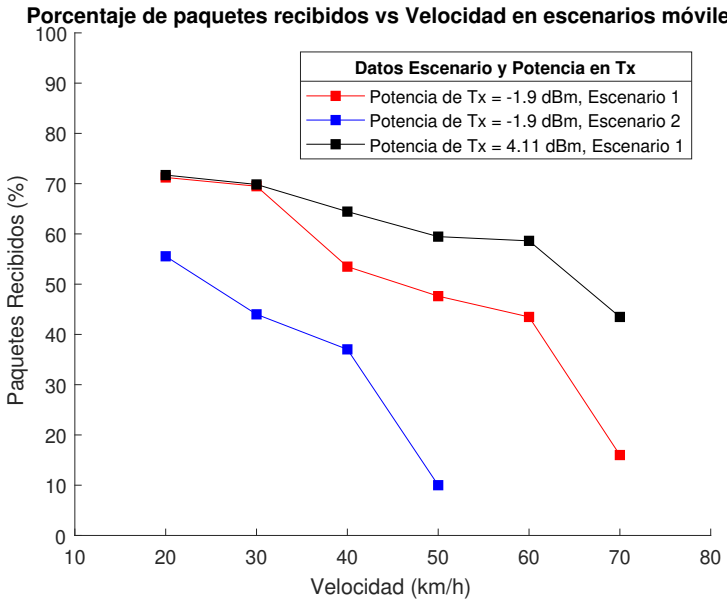


Figura 6.14: Porcentaje de paquetes recibidos en función de la velocidad

6.2.3.2. Análisis del retardo

El retardo obtenido en los escenarios móviles se muestra en la Figura 6.15. En el `Escenario 1` a una potencia de transmisión de 4.11 dBm, el retardo se sitúa por debajo de los 2 segundos. Una vez que la velocidad del vehículo aumenta el retardo es menor. Al transmitir a una potencia de -1.9 dBm, el retardo incrementa su valor en comparación con la transmisión a una mayor potencia. El retardo se encuentra entre los 2 y 2.5 segundos. El retardo decrece a medida que la velocidad del vehículo aumenta.

EL retardo disminuye a medida que la velocidad de la OBU aumenta. En el `Escenario 1` (Figura 6.9) la geografía del lugar permite que la trayectoria de la señal tenga menos distancia que recorrer. Al utilizar una potencia de -1.9 dBm cuando se moviliza la OBU a 70 Km/h, el retardo aumenta debido a que existe un vehículo que se moviliza por delante. Lo mismo sucede cuando se transmite la señal a 4.11 dBm y la OBU se moviliza a 60 Km/h.

En el `Escenario 2`, el retardo se comporta de forma contraria. Mientras la velocidad del vehículo aumenta el retardo tiende a incrementarse. A una velocidad de 20 Km/h el retardo se encuentra entre 1 y 1.5 segundos. A una velocidad de 50 Km/h, velocidad máxima en el `Escenario 2`, el retardo alcanza un valor cercano a los 2 segundos. El aumento del retardo en el `Escenario 2` móvil, es una consecuencia de la geografía de este escenario (Figura 6.11), las antenas de la OBU y de la RSU no cuentan con una linea de vista desde el inicio de la transmisión por la pendiente y curva que presenta la vía.

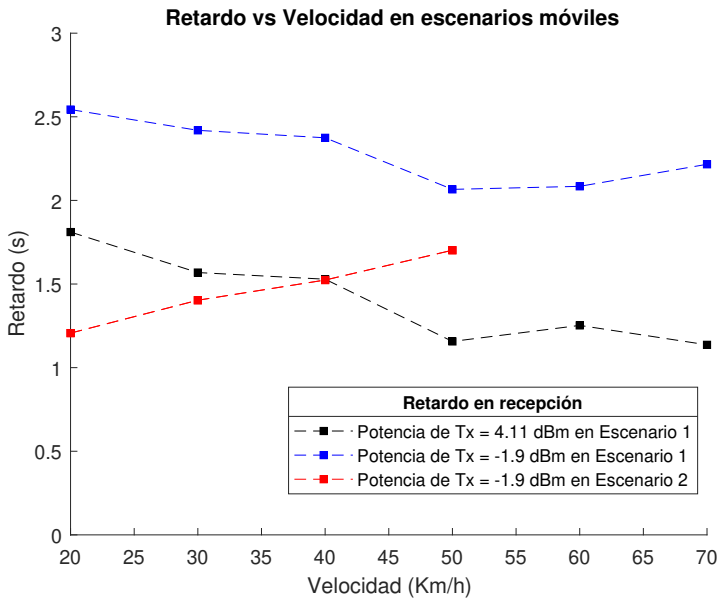


Figura 6.15: Retardo en función de la velocidad

6.3. Análisis del throughput

Al *throughput* se lo define como la tasa de transferencia de información efectiva sobre un canal de comunicación y es medido en el receptor. La tasa de transferencia total es conocida como *data rate* y se mide desde la fuente de la señal, es decir desde el transmisor. El *data rate* es medido en bits por segundo (bps) o en paquetes por segundo (p/s). Los resultados son presentados usando el *throughput* normalizado. Este hace referencia al *data rate* transmitido únicamente de la carga útil sobre el *data rate* máximo teórico en el medio, y es un valor adimensional con valores entre 0 y 1. De ahora en adelante el *throughput* normalizado se lo denominará únicamente como *throughput*. De acuerdo con las definiciones expuestas, se realiza el análisis del *throughput* para los escenarios implementados (estáticos y en movimiento). Los parámetros que se toman en cuenta en los experimentos son: esquemas de modulación, potencia en transmisión, distancia y velocidad.

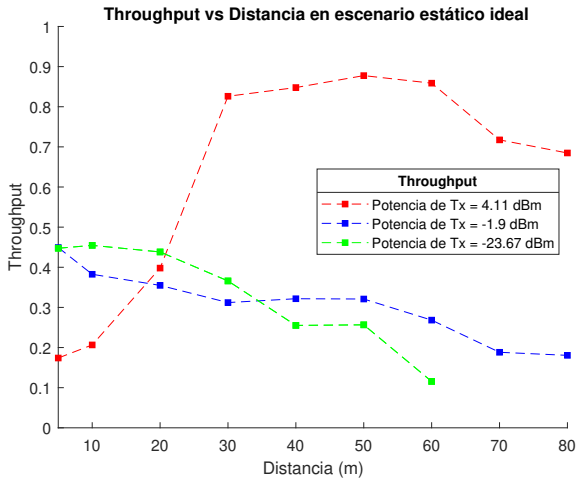
En cada transmisión que se realiza, se envía el mismo número de paquetes. Los paquetes que se envían tienen un tamaño fijo de 720 bytes (5760 bits) de los cuales 679 bytes son de datos, el resto de bytes corresponden a la información de los diferentes campos en la capa PHY y en la capa MAC. La carga útil corresponde a múltiples copias del estado del semáforo, el mismo que está compuesto por un número y una letra. El estado se repite varias veces para aumentar el volumen de información y completar los 679 bytes.

6.3.1. Throughput en el escenario ideal

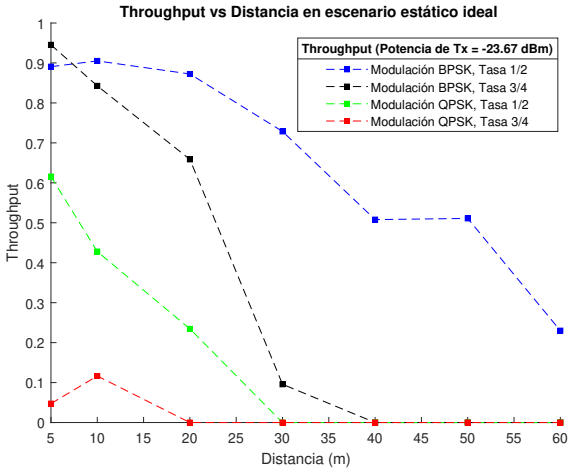
En la Figura 6.16 se encuentran los resultados de las pruebas realizadas en el escenario ideal. En la Figura 6.16b se muestra el análisis del *throughput* para dos modulaciones (BPSK y QPSK) a diferentes tasas de transmisión. La potencia de transmisión se mantiene en -23.67 dBm, con esta potencia se alcanza una cobertura de 60 metros. El *throughput* que mejor comportamiento presenta se da con la modulación BPSK 1/2. Con esta modulación a los 20 metros, se presenta un mayor *throughput* (por encima de 0.9) en comparación a las demás modulaciones. A medida que supera esta distancia, el *throughput* disminuye hasta alcanzar un valor

aproximado de 0.2 a los 60 metros. En la Figura 6.16b se observa que para las modulaciones restantes, existe una transferencia de datos hasta los 30 metros. Desde los 30 metros en adelante el valor de *throughput* es cero.

En la Figura 6.16a se analiza el *throughput* al enviar la señal OFDM a tres diferentes potencias y variando la distancia entre la RSU y la OBU. El *data rate* tiene un valor de 11800 bps. Al transmitir a potencias de -23.67 dBm y -1.9 dBm, el *throughput* alcanza un 0.5 a una distancia de 5 metros. Después empieza a descender conforme aumenta la distancia. Al transmitir a una potencia de -23.67 dBm, solo se cubre una distancia de 60 metros. El *throughput* a los 60 metros alcanza un valor por debajo del 0.2. Al transmitir con una potencia de -1.9 dBm, se cubre una distancia de 80 metros. Con esta potencia se obtiene un *throughput* de 0.2. Al transmitir a una potencia de 4.11 dBm, el *throughput* es de 0.2 a una distancia de 5 metros, aumenta hasta llegar a un valor cercano a 0.8 a los 30 metros. El *throughput* se mantiene alrededor de este porcentaje hasta una distancia de 60 metros. Después desciende hasta alcanzar un valor de 0.7 a una distancia de 80 metros. Este valor de *throughput* en recepción utilizando 4.11 dBm, es mayor al 0.2 que se obtiene con las potencias de -23.67 dBm y -1.9 dBm.



(a) Throughput al transmitir a diferentes potencias en el escenario ideal



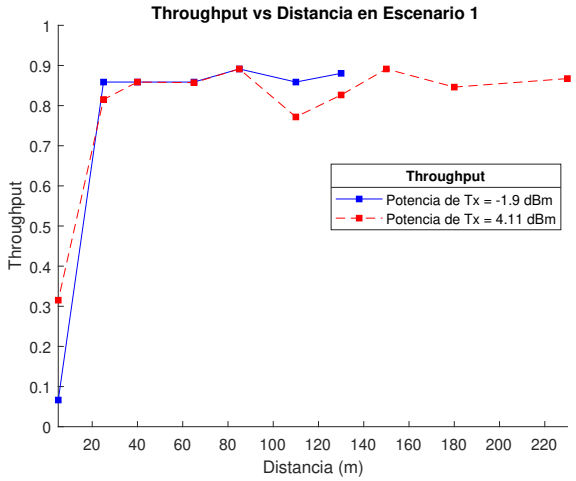
(b) Throughput al variar distancia y modulación en el escenario estático ideal

Figura 6.16: Throughput en recepción en el escenario ideal

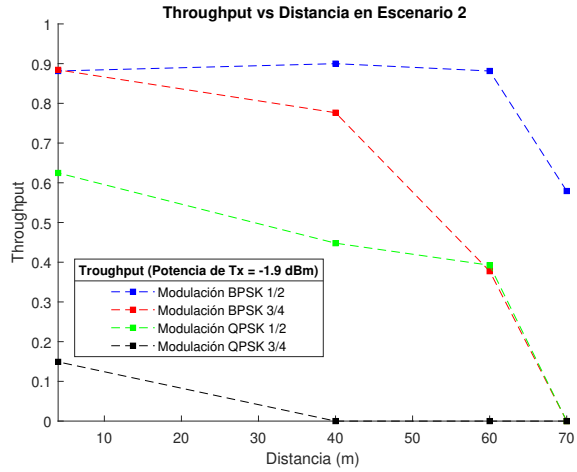
6.3.2. Throughput en los escenarios estáticos

Los resultados del *throughput* con referencia a los escenarios estáticos se muestran en la Figura 6.17. En la Figura 6.17a se observa los resultados de *throughput* al variar potencia y distancia en el `Escenario 1`. El *data rate* alcanza un valor cercano a los 12 Kbps. Al transmitir a una potencia de -1.9 dBm y 4.11 dBm existe un *throughput* de 0.1 y de 0.3 respectivamente, para una distancia de 5 metros. Al incrementar la distancia el valor de *throughput* tiende a aumentar, logrando estabilizarse en valores cercanos a 0.9. El punto crítico se encuentra a una distancia de 5 metros en donde el *throughput* está por debajo en comparación al *data rate* transmitido.

En la Figura 6.17b se muestran los resultados del *throughput* en el `Escenario 2`. Los valores del *throughput* están en función de la distancia y del esquema modulación. El *data rate* en transmisión tiene un valor de 11 Kbps. El esquema de modulación que mejor se comporta en recepción es BPSK 1/2. El *throughput* para esta modulación alcanza un 0.9. El *throughput* en recepción con una modulación BPSK 3/4 empieza con un valor cercano a 0.9, pero su rendimiento desciende a medida que la distancia se incrementa. Lo mismo ocurre con la modulación QPSK tasa 1/2 y tasa 3/4 cuyo valor de *throughput* inicial está en un 0.6 e inferior a 0.2 respectivamente. Igualmente disminuye y pierde rendimiento conforme aumenta la distancia.



(a) Throughput en función de la distancia y la potencia de transmisión en el Escenario 1



(b) Throughput en función de la distancia y modulación en el Escenario 2

Figura 6.17: Throughput en escenarios estáticos reales

6.3.3. Throughput en los escenarios móviles

Los resultados obtenidos del *throughput* en los escenarios móviles se observan en la Figura 6.18. El *data rate* en transmisión tiene un valor cercano a los 12 Kbps. Al analizar el `Escenario 1` al transmitir a una potencia de -1.9 dBm y 4.11 dBm, el *throughput* es similar para las velocidades menores a 60 Km/h cuyo valor está entre un 0.75 y 0.8. Cuando el vehículo circula a una velocidad de 70 Km/h existe una diferencia en *throughput* ya que, este disminuye a un 0.2 cuando la transmisión se da a una potencia de -1.9 dBm.

Para el `Escenario 2` el *throughput* es de 0.7 a una velocidad de 20 Km/h, este es el *throughput* máximo alcanzado en el `Escenario 2`. Entre las velocidades de 30 y 50 Km/h el *throughput* fluctúa entre 0.1 y 0.3. Este valor pequeño de *throughput* se debe a que, en este escenario la calle es curva y tiene una pendiente pronunciada. Por ende la alineación de antenas y la línea de vista es un punto importante a tomar cuenta.

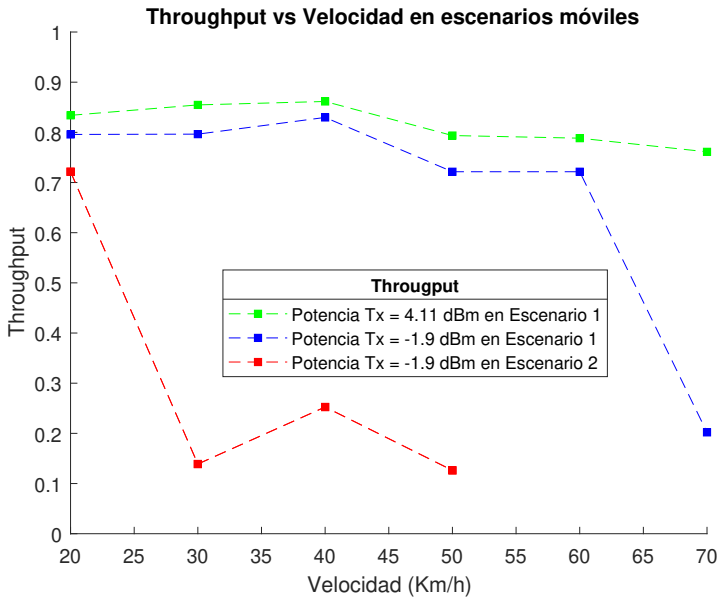


Figura 6.18: Throughput en escenarios móviles reales

6.4. Conclusiones

Al variar la potencia de transmisión en el escenario ideal, manteniendo la modulación BPSK 1/2, se observó que, a una potencia de -23.67 dBm se consigue un porcentaje de paquetes recibidos del 80 % a distancias menores a 20 metros. Pero con esta potencia en transmisión la cobertura máxima es de 60 metros. El porcentaje de paquetes recibidos disminuye al aumentar la distancia de cobertura. A una potencia en transmisión de -1.9 dBm, se obtiene un 80 % en paquetes recibidos a una distancia de 5 metros. Pero la cobertura ahora alcanza los 80 metros. Es decir con el aumento de potencia se logra cubrir una mayor distancia entre la RSU y la OBU. El porcentaje de paquetes recibidos sigue la misma tendencia, disminuir a medida que aumenta la distancia. El retardo a una potencia de -1.9 dBm se encuentra por debajo de los 2.5 segundos mientras que para una potencia de -23.67 dBm el retardo es superior, toma valores entre los 2.5 y 3 segundos.

Al usar la modulación BPSK 1/2 y la potencia de 4.11 dBm, el HackRF entra en un estado de saturación. Por lo que la recepción de paquetes a distancias menores a los 30 metros está por debajo del 40 %, lo que contrasta con lo analizado con las dos potencias anteriores (-23.67 dBm y -1.9 dBm). Esto se debe a que el dispositivo

entra en saturación y no resulta eficiente a distancias menores a 30 metros. Pero al llegar a 30 metros o distancias superiores, el porcentaje de paquetes aumenta a un 70 % en paquetes recibidos a una distancia de 80 metros. El retardo con esta potencia de transmisión es inferior a 1.5 segundos. Por lo que usar la modulación BPSK 1/2 y la potencia de 4.11 dBm para transmisión resulta ser la más eficiente.

Dentro del `Escenario 1` estático se envió la señal OFDM a dos diferentes potencias -1.9 dBm y 4.11 dBm. Estas dos potencias tienen mayor distancia de cobertura con respecto al escenario ideal. En el escenario ideal la distancia de cobertura estaba limitada a 80 metros debido a la geografía del terreno. Con la potencia de -1.9 dBm se alcanza una cobertura de 130 metros, mientras que con la segunda potencia una cobertura de 230 metros. En cuanto al porcentaje de paquetes recibidos a una potencia de -1.9 dBm, se obtiene un valor entre el 70 % y 80 % en los 130 metros de cobertura. Con una potencia de 4.11 dBm el HackRF se satura a una distancia de 5 metros. Por lo que los paquetes recibidos están por debajo del 30 %. Conforme se supera los 5 metros de distancia, el porcentaje de paquetes se mantiene entre el 70 % y 80 %, en los 230 metros de cobertura. En cuanto al retardo, cuando se transmite a una potencia de 4.11 dBm se encuentra por debajo de los 3 segundos, alcanzando un mínimo de 1.8 segundos a una distancia de 230 metros. Con la potencia de -1.9 dBm el retardo es superior a los 3 segundos.

En las pruebas en movimiento, se ve una curva decreciente en el porcentaje de paquetes recibidos al aumentar la velocidad de la OBU, las mediciones se realizaron con una velocidad desde los 20 km/h hasta los 70 km/h. Con una menor velocidad, el sistema es más estable y el enlace es más eficiente, el tiempo que le toma llegar a los paquetes desde la RSU hasta el OBU, disminuye alrededor de medio segundo.

Hay que tomar en cuenta que existen factores de rebote y reflexión de la señal. Estos factores son beneficiosos al enlace y factores como obstrucción, efecto Doopler, desvanecimiento de canal, no ayudan a un enlace óptimo. Además en el análisis del retardo, tanto en los escenarios reales como en el escenario ideal, se toma en cuenta el tiempo de procesamiento y visualización de los resultados.

Capítulo 7

Conclusiones, recomendaciones y trabajos futuros

7.1. Conclusiones

En el presente trabajo, el estándar IEEE 802.11p está implementado en 2 programas separados, un programa para el transmisor, el cual usa la RSU y otro programa que utiliza el receptor, la OBU. Dentro del programa del transmisor se utiliza un programa jerárquico para enlazar la capa PHY con la capa MAC. Los bloques empleados permiten formar los campos necesarios, en las diferentes capas, además de realizar los procesos necesarios para formar y compactar la señal a enviar. En el desarrollo de los programas se utilizó diversos códigos y librerías. Estas están programadas en Python y en C++. En cuanto al programa del receptor, no utiliza un programa jerárquico. Todo el proceso para obtener la información está realizado en un solo programa. Este programa se enfoca en obtener la información útil, es decir, el estado del semáforo. El proceso de recepción consta de la detección de la señal, la decodificación de la señal y finalmente, la extracción de la carga útil.

Dentro del desarrollo de este proyecto, se evidenció que las capacidades de cómputo para el dispositivo HackRF ONE son limitadas. Al momento de utilizar un esquema de modulación superior a QAM el transmisor logra enviar con gran dificultad la señal OFDM luego de un periodo de tiempo prolongado. En pruebas de campo al utilizar una modulación superior a QAM, la señal no es construida de forma correcta y al ser recibida por la OBU no se reconstruye correctamente. Para la estimación de canal, en todas las pruebas realizadas se utilizó el esquema de mínimos cuadrados. Este algoritmo se lo elige por defecto, a pesar de haber tres algoritmos más. Se consideró trabajar solo con el algoritmo mencionado, porque es el que menos recursos computacionales utiliza, así se demuestra en [67].

Se diseñó una aplicación que envía mensajes *broadcast* para la comunicación entre la RSU y la OBU. Los mensajes contienen el estado de un semáforo de tránsito y son emitidos desde la RSU. Para que el usuario utilice la información recibida se trabajó en varias pruebas. Para la interfaz de usuario se optó en realizar una interfaz gráfica en donde se observa el semáforo y su estado actual de acuerdo a la información que se reciba en la OBU.

La configuración usada en la RSU es el dispositivo HackRF ONE en cascada con el amplificador lineal. Al realizar la caracterización con esta configuración en el rango de frecuencias de 1GHz a 6GHz, se observó que la RSU alcanza una mayor cobertura con un mayor número de paquetes recibidos en la frecuencia de 2.5 GHz. En esta frecuencia se alcanzó una potencia de transmisión de 4 dBm. En la frecuencia de 5.890 GHz (frecuencia

del estándar IEEE 802.11p) se alcanzó una potencia de transmisión de -5 dBm. Con esta potencia no se superó un metro de cobertura entre la RSU y la OBU. Aunque la frecuencia de trabajo del estándar IEEE 802.11p es en la banda de 5.890 GHz, debido a las limitaciones del HackRF ONE en esta frecuencia, se usó la banda de 2.500 GHz para realizar las pruebas.

En las pruebas de laboratorio, se usó la modulación BPSK 1/2 y al ancho de banda de 2 MHz. Con estos dos parámetros y junto a la frecuencia de 2.484 GHz, se observó que el HackRF ONE transmite la señal OFDM a una potencia constante y con un espectro bien definido. Cuando se usó otra modulación (por ejemplo 16 QAM) u otro ancho de banda (10 MHz), el HackRF ONE no realiza el procesamiento necesario para poner en marcha la transmisión.

Con la caracterización del HackRF ONE en recepción, se observó que el dispositivo al capturar potencias superiores a los -40 dBm entra en un estado de saturación. Este problema hace que el dispositivo no sea capaz de recibir paquetes desde el transmisor. Así mismo se identificó un rango de potencia que va de -65 dBm a -95 dBm, para que el HackRF ONE vuelva a funcionar como receptor. En este intervalo de potencia el porcentaje de paquetes recibidos aumenta progresivamente, llegando a un máximo de 80 %. Al superar los -95 dBm de potencia en recepción el porcentaje de paquetes recibidos empieza a descender.

Los escenarios usados para las pruebas de campo son un escenario ideal (estático) y un escenario real (en movimiento). En el escenario ideal al no tener obstáculos entre la RSU y la OBU, se alcanzó una potencia en recepción superior a -85 dBm a 80 metros de distancia, con una potencia en la RSU de 4.11 dBm. La potencia de recepción de -85 dBm aseguró que, al empezar las pruebas en los escenarios reales se supere sin dificultad la distancia de cobertura del escenario ideal (80 metros). En los escenarios reales se identificó a los vehículos (estos actúan como obstáculo entre la RSU y la OBU), la alineación entre antenas y la velocidad de la OBU como factores que influyen en la transmisión y recepción de la señal OFDM.

Para las pruebas de campo en transmisión y en recepción se varió parámetros como modulación y potencia de transmisión. En el escenario ideal se identificó a BPSK 1/2 como la modulación con el mayor porcentaje de paquetes recibidos (superior al 80 %), en distancias menores a los 20 metros. Con el uso de esta modulación, el porcentaje de paquetes recibidos disminuye a medida que aumenta la distancia. Se obtiene alrededor del 3 % a una distancia de 60 metros. Además se obtiene también un retardo por debajo de los 2 segundos.

Al transmitir con una potencia de 4.11 dBm en el escenario ideal, se observó una saturación por parte de la OBU en los primeros 20 metros. Por lo tanto el porcentaje de paquetes recibidos es inferior al 40 %. Con esta misma potencia a los 50 metros de distancia, el porcentaje de paquetes aumentó hasta un 80 %. A distancias de 20 metros o inferiores, el HackRF ONE no procesa los paquetes recibidos debido a que se encuentra en saturación. Para la aplicación que se desarrolló en este proyecto, es necesario cubrir un rango de cobertura de 80 metros del escenario ideal o una distancia superior, aunque el comportamiento a distancias cortas (20 metros) entre la RSU y la OBU no sea el esperado, debido a la poca cantidad de paquetes que se recibe.

Una de las potencias de transmisión usadas en el `Escenario 1` estático es de -1.9 dBm. Con esta potencia, se observó que la distancia de cobertura es de 130 metros entre la RSU y la OBU. En esta potencia se alcanzó un porcentaje de paquetes cercano al 70 % a los 5 metros de distancia. Con una potencia de 4.11 dBm se obtuvo una cobertura de 230 metros. Al usar esta potencia se superó las expectativas de cobertura que se tenía en un principio (80 metros en el escenario ideal). Con la potencia de transmisión de 4.11 dBm, se observó que el HackRF ONE en recepción se saturó a una distancia de 5 metros, por lo que se obtuvo un porcentaje de paquetes inferior al 30 %. Al usar las dos potencias de transmisión (-1.9 dBm y 4.11 dBm) el retardo presentó resultados diferentes. Con una potencia de 4.11 dBm el retardo está por debajo de los 3 segundos mientras que con una potencia -1.9 dBm, el retardo fue mayor a 3 segundos.

En el escenario en movimiento, cuando la OBU se movilizó entre los 20 Km/h y 30 Km/h, se obtuvo un mayor porcentaje de paquetes recibidos (70 %). Al superar los 30 km/h, el porcentaje de paquetes que llegan a la OBU disminuye. Con velocidades menores a los 60 Km/h se obtuvo un throughput estable y un retardo que se mantiene entre los 2 y 2.5 segundos.

7.2. Recomendaciones

Para una alineación más precisa con las antenas de los equipos, es conveniente obtener el patrón de radiación de la antena. Este es un dato importante que permite conocer las características de radicación en función de la potencia. En la aplicación implementada en este trabajo es necesario usar una antena directiva. De este modo la información llega únicamente al carril que se está utilizando y analizando. La señal no debería llegar después de que la OBU pasa la RSU que en este trabajo está montada dentro de un semáforo.

Para replicar estos experimentos, se recomienda realizar pruebas en laboratorio antes que las pruebas en campo. Es necesario comprobar que se cumplan todos los parámetros que establece el estándar a utilizar. Se debe revisar y comprobar los parámetros técnicos del dispositivo SDR por medio de pruebas de potencia y radiación para validar su funcionamiento adecuado. Los parámetros del amplificador a utilizar deben ser conocidos. Los parámetros más importantes son: voltaje mínimo y máximo de alimentación, ganancia a obtener según el nivel de voltaje de ingreso y la potencia mínima y máxima de alimentación. Estas recomendaciones evitarán que el dispositivo SDR se sature, se descomponga o se queme en el peor caso.

Las computadoras deben estar sincronizadas usando el protocolo Real Time Transport Protocol (RTP). Caso contrario las marcas de tiempo en el analizador de paquetes saldrán erróneas y no será posible el análisis de los resultados.

7.3. Trabajos Futuros

La implementación del estándar IEEE 802.11p en el HackRF ONE funciona de forma correcta. En este trabajo la cantidad de información que se ha transmitido ha sido mínima, por lo que para un trabajo futuro se propone enviar una mayor cantidad información, a más del estado del semáforo, añadir sensores como por ejemplo temperatura, humedad, velocidad, entre otros.

Las pruebas realizadas en el presente trabajo, han sido con una RSU y una OBU, para futuros trabajos se propone el estudio del estándar IEEE 802.11p sobre el dispositivo HackRF ONE, añadiendo una OBU (2 en total) y analizar en los escenarios móviles, realizando los siguientes experimentos: una OBU movilizándose detrás de otra, una OBU sobrepasando en velocidad a la otra y cuando las OBUs se muevan de forma simultánea y a la misma velocidad.

Anexo A

Instalación de Software

A.1. Instalación de GNU Radio

Para la instalación de GNU Radio se utilizó una máquina virtual con Ubuntu 18.04. Además, se instaló el *software* necesario para el transmisor y receptor. Se empieza por la instalación de GNU Radio. La instalación se realiza usando el comando mostrado en el Listado A.1.

```
sudo apt install gnuradio
```

Listado A.1: Instalación gnuradio

Como siguiente paquete a instalar está el bloque RTL SDR. Este bloque es usado por GNU Radio para la lectura de los diferentes *hardware* SDR, no es necesario especificar el modelo de hardware a usar. El bloque se instala usando el comando del Listado A.2.

```
sudo apt install gr-osmosdr
```

Listado A.2: Instalación gr-oscomosdr

Una vez instalado GNU Radio 3.7 y el bloque RTL SDR se accede a GNU Radio con el comando del Listado A.3.

```
sudo gnuradio-companion
```

Listado A.3: Ejecución gnuradio

Se procede a instalar la herramienta *swig*. Su instalación se realiza usando el comando del Listado A.4. *swig* es un compilador que facilita la integración del lenguaje C y C++ con otros lenguajes de programación como Perl, Ruby, Python, Java, etc.

```
sudo apt-get install swig
```

Listado A.4: Instalación swig

A.2. Instalación de Bloques IEEE 802.11

Los bloques necesarios para implementar el estándar IEEE 802.11p se encuentran en un repositorio en `GitHub`. Para clonar el repositorio en el directorio *home*, se utiliza la herramienta `git`. La instalación del repositorio se realiza usando el comando mostrado en el Listado A.5.

```
git clone https://github.com/IBM/dsrc.git
```

Listado A.5: Instalación del repositorio dsrc

A continuación se instala los bloques del repositorio `gr-ieee802-11`. Dentro del directorio *build*, se coloca el archivo de *cmake-3.7.2.tar.gz* previamente descargado desde [63]. La instalación de los bloques se realiza usando la lista de comandos mostrado en el Listado A.6.

```
cd dsrc
cd gr-ieee802-11
mkdir build
cd build
tar -xvzf cmake-3.7.2.tar.gz
cd cmake-3.7.2/
sudo ./configure
sudo make
sudo make install
cmake --version
sudo cmake ..
sudo make
sudo make install
sudo ldconfig
```

Listado A.6: Instalación bloques gr-ieee802-11

Finalmente se procede a instalar los bloques del repositorio *gr-foo*. Para la instalación del repositorio se usan los comandos del Listado A.7.

```
    cd dsrc
    cd gr-foo
    mkdir build
    cd build
    sudo cmake ..
    sudo make
    sudo make install
    sudo ldconfig
```

Listado A.7: Instalación bloques gr-foo

Anexo B

Desarrollo de la aplicación

B.1. Descripción de la aplicación

La aplicación desarrollada tiene como finalidad compartir el estado del semáforo desde la RSU hacia la OBU. Para ello, la aplicación se divide en dos partes. La primera parte consiste en adquirir el estado del semáforo e integrarlo al programa en transmisión como carga útil. Se utiliza un Arduino UNO, encargado de controlar el semáforo en tiempo de encendido/apagado de cada panel de luces. Estas luces contemplan un dato por estado (color de los paneles), el mismo que es enviado desde Arduino hacia la máquina virtual de forma serial. El estado del semáforo es guardado en un archivo `txt` para su lectura por el programa transmisor.

La segunda parte de la aplicación es la recepción del dato por parte de la OBU. Luego de pasar por la capa PHY y por la capa MAC, el estado del semáforo es guardado temporalmente en un archivo `txt`, para posteriormente ser leído por Node Red. Mediante este programa el estado del semáforo se mostrará mediante una interfaz de usuario.

B.2. Manejo del semáforo por parte de la RSU mediante Python y Arduino.

En la máquina virtual que se ejecuta la RSU, se procede a crear un algoritmo mediante `Python-3`. En el algoritmo se usan dos librerías: `time` y `Pyserial`, esta última permite la comunicación serial con la placa de Arduino UNO. La instalación de la librería `Pyserial` se lo realiza mediante el comando que se muestra en el Listado B.1.

```
python3 -m pip install PySerial
```

Listado B.1: Instalación PySerial

En la Figura B.1 se muestra el algoritmo creado para la comunicación serial entre Python y Arduino. Este algoritmo parte de la variable `arduino`, que contiene a la clase `Serial`. Esta clase puede recibir varios parámetros, pero los más importantes son los siguientes:

- El puerto Universal Serial Bus (USB) al cual está conectada la placa de Arduino. El nombre por defecto que lleva el puerto es `/dev/ttyACM0`
- La velocidad en baudios, para este caso se usa 9600.

En la variable *data* se usa la función `read` para realizar la lectura del dato que ingrese por el puerto serial. El Arduino envía datos en caracteres ASCII. Para decodificar este dato se usa la función `decode`, con la cual se transformará en un valor entero. Este dato se lo guarda en la variable `datad`. El valor de esta variable se la escribe dentro del archivo `datos.txt`.

Con el proceso explicado se lee solo el primer dato que llegue desde el Arduino, pero se mantiene abierta la comunicación serial entre la placa y el ordenador. La función `arduino.close()` cierra la comunicación serial y sirve para terminar la ejecución del programa. Mientras esta función no se ejecute, la comunicación serial se mantiene abierta.

Como siguiente paso se guarda el valor de `datad` leído anteriormente en la variable `datad1`. Se lee un nuevo dato que ingresa por el puerto serial y se lo guarda en `data`. El valor de esta variable es un caracter ASCII, se lo decodifica mediante la función *decode*. Este dato decodificado se lo guarda en la variable `datad`. A continuación, se usa una sentencia `if`, la cual entra en ejecución cuando la variable `datad` tiene un valor diferente al almacenado en la variable `data1`. Si los valores son diferentes, el dato guardado en *datad* se sobre escribe en el archivo `datos.txt`. Este bucle se mantiene hasta la ejecución de la función `arduino.close()`. Con esta función, se cierra la conexión serial entre Python y la placa de Arduino.

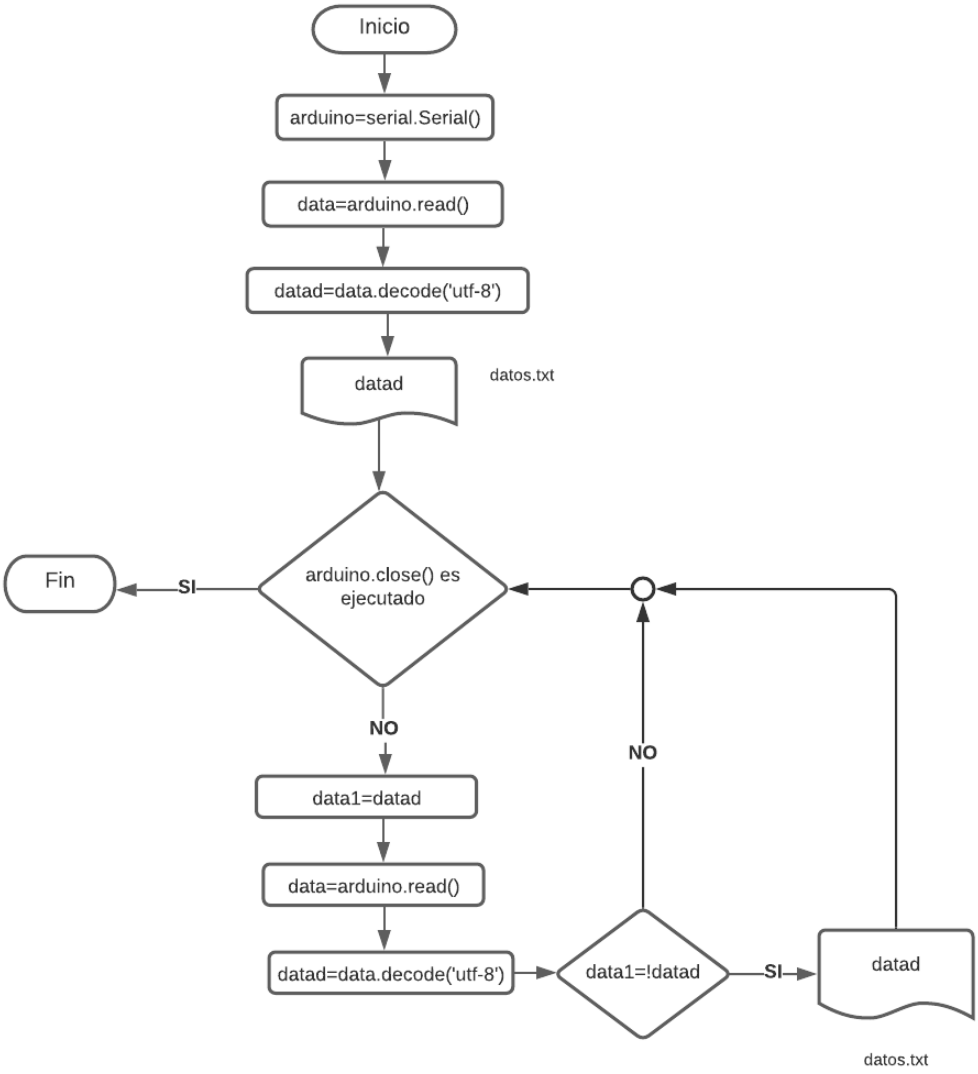


Figura B.1: Algoritmo para la comunicación serial entre Python y Arduino

En la Figura B.2 se muestra el algoritmo para el control del semáforo mediante Arduino. Con este algoritmo también se envía el estado del semáforo hacia el ordenador, usando el puerto serial.

Se define 3 variables para identificar los estados del semáforo, estos valores son los que se envían por el puerto serial hacia el ordenador. La variable `val1` toma el valor de 1 y representa el estado rojo en el semáforo, `val2` toma el valor de 2 y representa el estado amarillo y la variable `val3` toma el valor de 3 y representa el estado verde. Se define los pines 2, 4 y 6 del Arduino como pines de salida y se los inicializa de la siguiente manera: `PIN2` en alto (`HIGH`), `PIN4` y `PIN6` en bajo (`LOW`).

A continuación se ejecuta el primer bucle `for` por 10 ocasiones, con un retardo de un segundo por cada ejecución. Dentro de este bucle se envía por puerto serial el valor de la variable `val1`(estado rojo del semáforo).

Terminada la ejecución del primer bucle `for`, se envía una señal de encendido (`HIGH`) al `PIN4`. Este pin controla el estado amarillo del semáforo.

Seguidamente se encuentra el segundo bucle *for*. Este bucle se ejecuta en tres ocasiones con un retardo de un

segundo por cada ejecución. Dentro de este bucle se envía por puerto serial el valor de `val2` (estado amarillo del semáforo).

Como siguiente paso, se envía una señal de apagado (`LOW`) a los pines 2 y 4, mientras que al pin 6 se envía una señal de encendido (`HIGH`). Con estas señales se apagan las luces rojo y amarillo del semáforo y se enciende la verde.

Con el tercer bucle *for* que se ejecuta 10 veces con un retardo de un segundo por cada ejecución, se envía por puerto serial el valor de la variable `val3` (estado verde del semáforo). Al final de este bucle, el algoritmo regresa a la parte donde se define los estados iniciales de los pines (función `loop`).

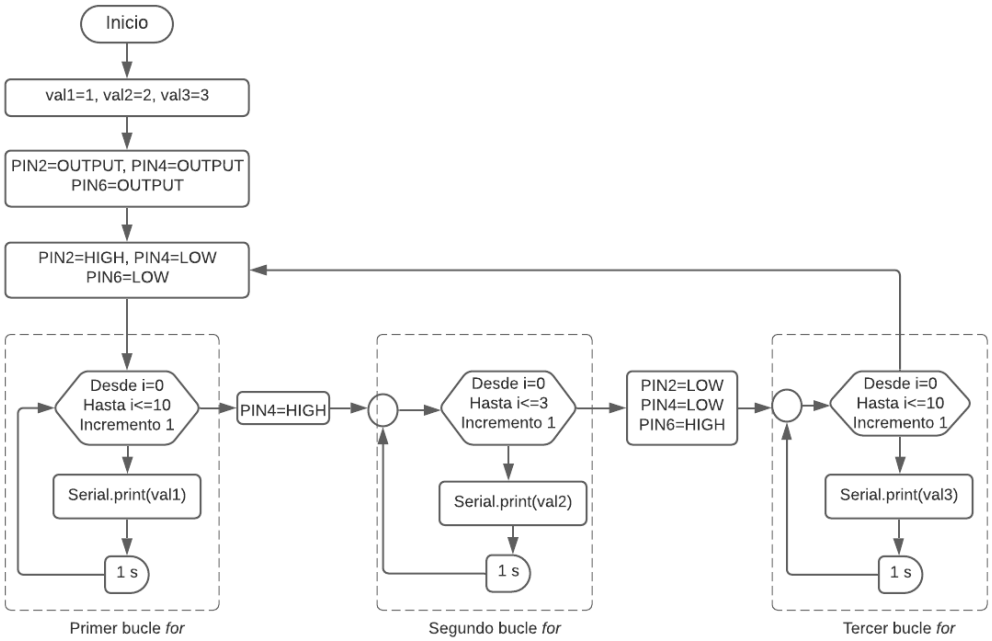


Figura B.2: Algoritmo para el control del semáforo mediante Arduino

B.3. Interfaz gráfica de usuario para la OBU

Instalado el software Node Red, se procede a la programación por bloques. Para acceder a el área de trabajo se ingresa a la dirección por defecto: `https://localhost:1800`. El diagrama general se muestra en la Figura: B.3. En donde se lee el archivo que va a contener la carga útil, es decir el estado del semáforo, este dato es enviado desde el RSU hacia el OBU.

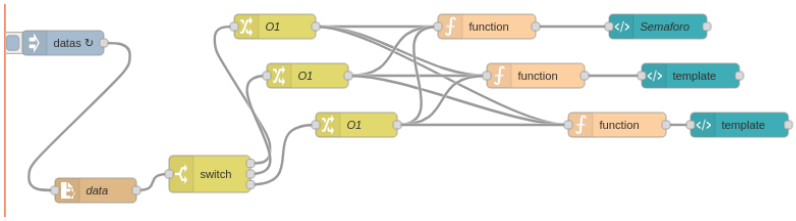


Figura B.3: Diagrama de bloques para la interfaz de usuario

Se selecciona que cada medio segundo lea el archivo. Se cuenta con un bloque que permite validar el tipo de dato, si el semáforo se encuentra en el estado rojo, el dato a enviar es A1. En estado amarillo, el mensaje es A2. Si el semáforo esta en verde, el dato es A3. Una vez validado el estado del semáforo, se pasa a una función que nos permitirá realizar el circulo y agregar el color deseado, como se ve en la Figura B.4. Esto se realiza para cada estado del semáforo, es decir para el color de luces: rojo, amarillo y verde. Una vez diseñado se envía a mostrar en la interfaz. Se da clic en desarrollar e se ingresa a la dirección `https://localhost:1800/ui`.

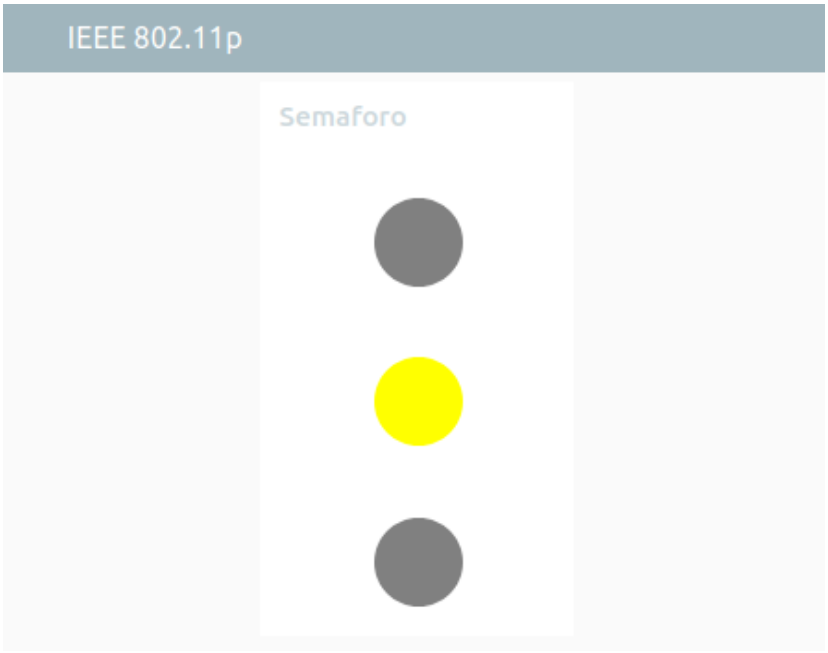


Figura B.4: Interfaz de usuario

.

Anexo C

Descripciones de software y hardware

C.1. Software GNU Radio

GNU Radio es un kit de herramientas de desarrollo de *software* gratuito y de código abierto que proporciona bloques de procesamiento de señales para implementar radios de *software*. Se puede usar con *hardware* de RF externo de bajo costo. GNU Radio permite utilizar radios definidas por *software* con o sin *hardware* en un entorno de simulación. Es ampliamente utilizado en investigación, industria, academia, gobierno y entornos de aficionados para apoyar al desarrollo de comunicaciones inalámbricas como los sistemas de radio del mundo real. [11]

C.1.1. Tipo de datos

GNU Radio permite trabajar con diversos tipos de datos y asigna un color característico a cada uno. Los datos serán usados según la necesidad o el requerimiento del usuario o por definición propia del bloque a utilizar. Entre los tipos de datos están los siguientes: flotantes complejos, complejos enteros, flotantes, enteros, bits sin empaquetar, mensajes asíncronos, bus y *wildcard*. En la Figura C.1 se observa cada uno de los tipos de datos con el color asignado. [68]

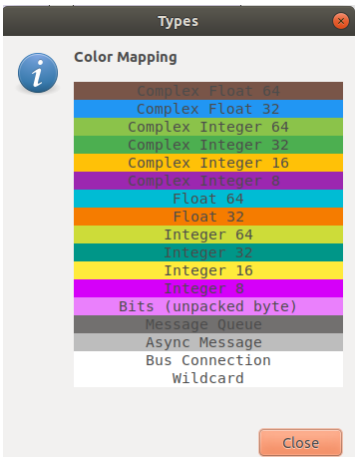


Figura C.1: Datos admitidos en GNU Radio

C.1.2. Programación

GNU Radio posibilita la programación de bloques y módulos desde su interfaz principal conjuntamente con un editor de código mediante ***python block*** y ***python module***. El bloque y el módulo se muestran en la Figura C.2.

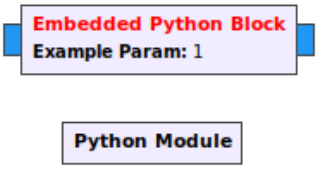


Figura C.2: Bloque y Módulo Python

C.1.2.1. Bloque Python

Permite crear bloques personalizados, programar las características que el usuario desea mediante código python. En la Figura C.3 se observa las características del bloque y el código por defecto que se genera, este bloque contiene un parámetro de entrada y uno de salida en formato Complex 64 y realiza la multiplicación del valor de entrada por una constante que ingrese en el bloque [69].

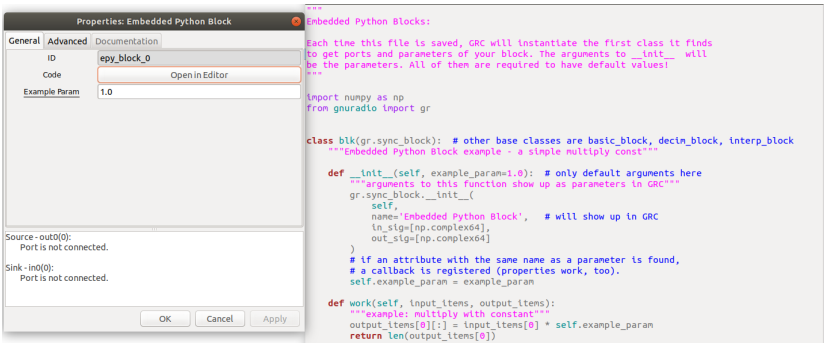


Figura C.3: Python Block

C.1.2.2. Módulo Python

En la Figura C.4 muestra el módulo python, el cual permite codificar de forma accesible bloques utilizando los identificadores del bloque, en esta figura el código se muestra vacío por defecto para la programación del mismo. Además se puede establecer parámetros de otros bloques en el diagrama de flujo [70].

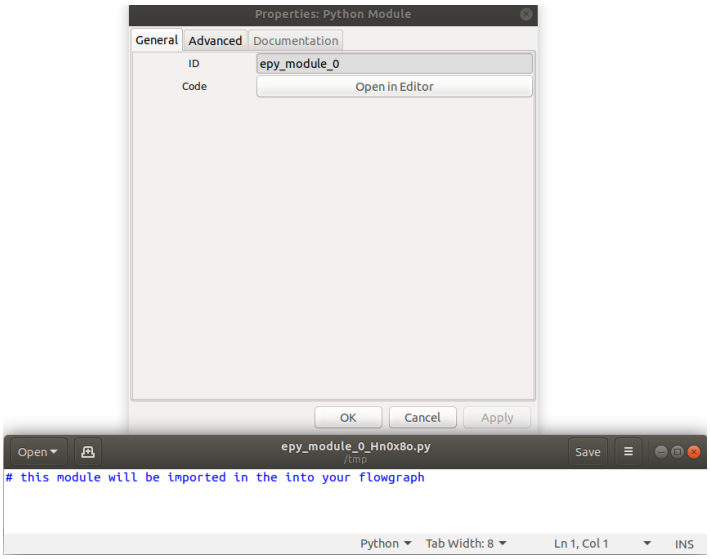


Figura C.4: Python Module

C.1.3. Visualizaciones de Resultados

Dentro de GNU Radio se desea visualizar gráficas de las señales que estamos adquiriendo u obteniendo. En GNU Radio se puede observar los resultados en dos tipos de interfaces: QT GUI y WX GUI. Cabe recalcar que, al elegir un tipo de interfaz, se debe escoger los bloques asociados a esta interfaz [71].

C.1.3.1. QT GUI

Estas herramientas están basadas en Python y consumen menos recursos en procesamiento. Esta opción estará habilitada dentro del bloque *Options* en el campo *Generate Options*. Existen varios entornos dentro de este bloque como los siguientes:

- ***QT GUI Frequency Sink:*** Con este bloque se puede visualizar las señales en el dominio de la frecuencia, se debe agregar el bloque y configurar el tipo de dato de entrada.
- ***QT GUI Time Sink:*** Con este bloque se puede visualizar el comportamiento en tiempo de la señal de entrada.
- ***QT GUI Constellation Sink:*** Permite observar las constelaciones de una señal modulada digitalmente.
- ***QT GUI Sink:*** Esta herramienta permite observar las 4 herramientas en una sola ventana con varias pestañas en la interfaz del usuario. Disminuye el procesamiento pues cada herramienta que no se utiliza se pone en pause mientras se utilice alguna.

C.1.3.2. WX GUI

Estas herramientas están basadas en C++ y consumen más recursos de procesamiento que las herramientas QT GUI. Si se quiere usar la opción WX GUI se habilitará en el bloque *Options* dentro del campo *Generate*

Options. Se reemplazará la opción QT GUI por WX GUI.

- ***WX GUI Scope Sink:*** Permite observar el comportamiento de las señales en el dominio del tiempo.
- ***WX GUI FFT Sink:*** Permite observar el dominio de la frecuencia de las señales, se observa el espectro de frecuencia.
- ***WX GUI Constellation Sink:*** Es la representación gráfica de las señales complejas, útil para el estudio y visualización de modulaciones digitales.
- ***WX GUI Histo Sink:*** Sirve para mostrar un histograma.

C.1.3.3. Bloques Jerárquicos

Son bloques especiales en cuanto a la forma de ser construidos. En vez de estar definidos por código directamente, se construyen mediante un diagrama de bloques en GNU Radio Companion (GRC). Por tanto, su comportamiento depende de de otros bloques. Ayuda a mantener más ordenados los *flowgraphs*. Estos tipos de bloques permiten la simplificación de los diseños y liberar el espacio de trabajo de una gran cantidad de bloques dependiendo de la complejidad de la aplicación. [72]

C.2. HackRF One

Es un dispositivo fabricado por Great Scott Gadgets, es considerado como un equipo SDR que permite el envío y recepción de datos, diseñado para la experimentación y desarrollo de soluciones en el ámbito de las comunicaciones inalámbricas. No es vendido directamente desde el fabricante por lo que podemos adquirirlo a un precio que varía desde $ 300 hasta $ 400. El dispositivo HacRF One es capaz de transmitir o recibir señales en el rango de 1 MHz hasta los 6 GHz. Está diseñado para la tecnología SDR. HackRF One es un *hardware* libre que puede ser programado para su funcionamiento autónomo. Este dispositivo tiene la capacidad de muestreo de hasta 20 millones de muestras por segundo, 8 bits de muestras en cuadratura. Posee un conector SMA hembra para la antena, además de un conector SMA hembra para sincronización. La Figura C.5 [73] muestra el dispositivo HackRF One sin su carcasa protectora [74].

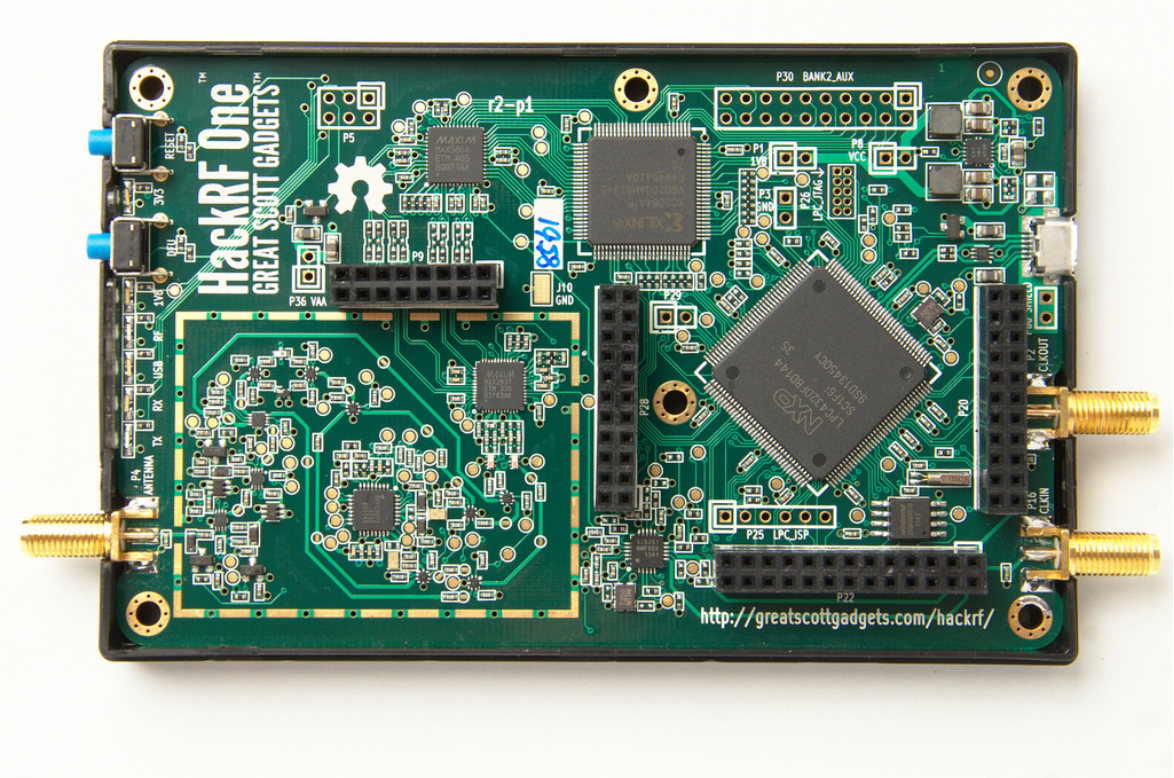


Figura C.5: HackRF One

Este dispositivo está compuesto por varios componentes mostrados en el diagrama de bloques en la Figura C.6.

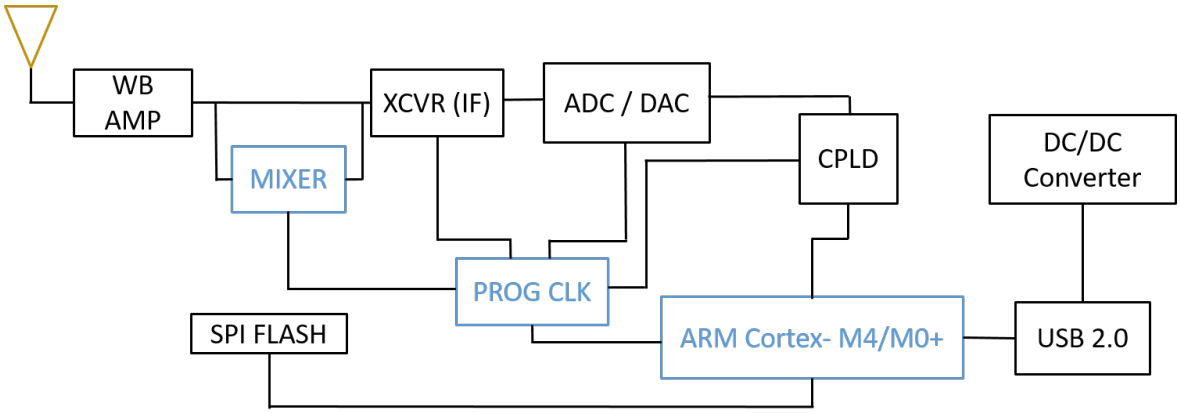


Figura C.6: Diagrama de Bloques HackRF One

- ***USB 2.0:*** Interfaz encargada de suministrar la alimentación al dispositivo.
- ***DC/DC Converter:*** Conversor reductor de voltaje(topología Buck) con dos salidas a tensiones diferentes que forma la fuente de alimentación de la placa.
- ***SPI FLASH:*** Memoria flash conectada al microcontrolador ADVANCED RISC MACHINE (ARM), permite la configuración del microcontrolador principal y de la CPLD, cada vez que se inicie HackRF One.

- ***PROG CLK:*** Integrado que a partir de un cristal como referencia permite generar varios relojes simultáneamente con una frecuencia independiente para cada reloj.
- ***RM Cortex-M4/M0+:*** Microcontrolador de 2 núcleos con arquitectura ARM
- ***ADC/DAC:*** Conversor compuesto por un ADC y un DAC, tienen la capacidad de trabajar simultáneamente.
- ***CPLD:*** Dispositivo lógico programable complejo, utilizado como un puente entre el conversor ADC/DAC y el procesador.
- ***XCVR IF:*** Transceptor que puede recibir o transmitir señales desde o hacia la banda de 2.15 a 2.75 GHz, utilizado como un conversor a frecuencia intermedia para trasladar a una frecuencia final usando el mezclador.
- ***MIXER:*** Mezclador utilizado para poder manejar señales fuera del rango de 2.15 a 2.75 GHz, desplazando la señal hasta una frecuencia 1 MHz y 6 GHz por medio de una portadora local.
- ***WB AMPL:*** Amplificador de banda ancha utilizado en la última etapa para transmitir y recibir [74].

C.3. Arduino Uno

Es una placa programable que utiliza el procesador ATmega 328. Como se muestra en la Figura C.7 se encuentra conformado por 14 pines de entrada/salida digital, de los cuales 6 pueden ser usados como Pulse-Width Modulation (PWM), 6 entradas analógicas, un cristal de 16 MHz, conexión USB, conector jack de alimentación, terminales para conexión In Chip Serial Programmer (ICSP) y un botón de *reset* [40]. En cuanto a su programación el ATmega328 viene preprogramado mediante un gestor de arranque (*bootloader*) permitiendo reprogramar el chip sin el uso de un programador de *hardware* externo. Usa el protocolo STK500 original para la comunicación. Su programación se lo realiza mediante el *software* IDE *open source* de Arduino que se lo puede descargar gratuitamente [40][41].

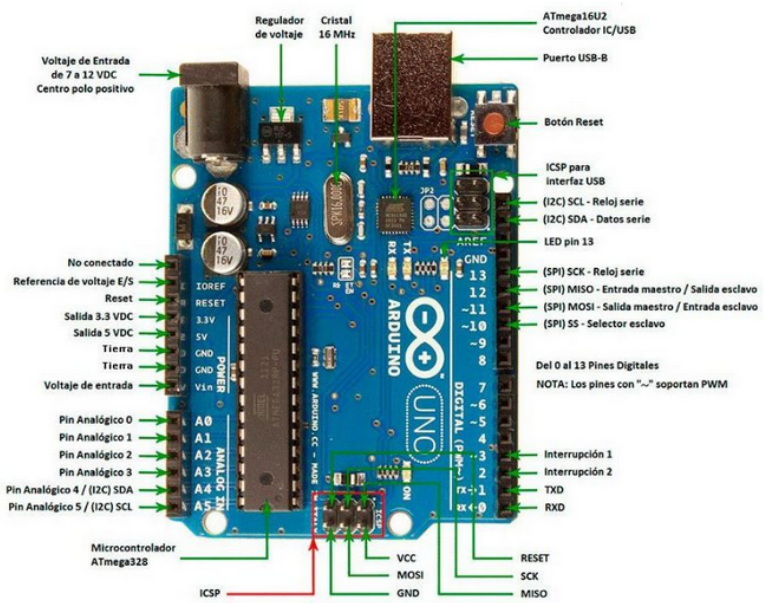


Figura C.7: Arduino Uno junto con su etiquetado [41]

A continuación se presenta un resumen de sus principales características:

- Microcontrolador ATmega328.
- Voltaje de entrada 7-12V.
- 14 pines digitales de I/O con 6 salidas PWM.
- 6 entradas analógicas.
- 32k de memoria Flash.
- Reloj de 16MHz de velocidad.
- Dimensiones; 68.6mm x 53.4mm.

C.4. Módulo de 4 relés para Arduino

Este módulo está formado por 4 relés que funcionan a 5V con la capacidad de manejar cargas de hasta 10A en 250V. Cada relé se encuentran aislados mediante octoacopladores de las entradas. Estas entradas cuentan con leds individuales que sirven como indicadores de estado [75].

Los componentes del módulo relé se observan en la Figura C.8 y se enumeran a continuación [75]:

1. Conectores de entrada IN1 a IN4, GND y Vcc.
2. Leds indicadores para los 4 relés.
3. Jumper selector para la alimentación de los relés.
4. Octoacopladores del tipo FL817C.
5. Diodos de protección.
6. Relés con bobinas a 5V, que pueden controlar cargas hasta 10A-250V.
7. Borneras formada por 3 contactos (común, normalmente abierto, normalmente cerrado).

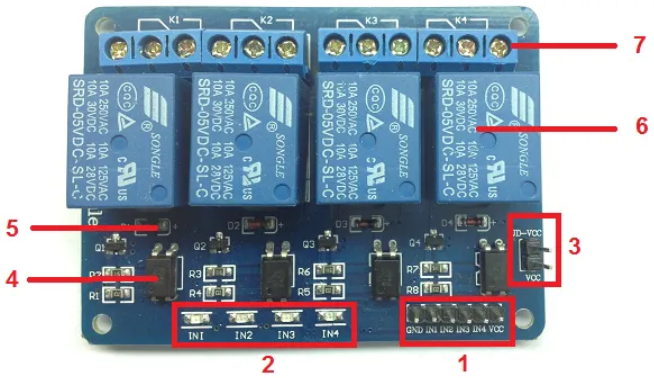


Figura C.8: Componentes del módulo relé [75]

C.5. Equipos de Medición

C.5.1. Analizador de espectros: HP-8593E

El Keysight 8593E es un analizador de espectro de microondas ofrece una amplia gama de rendimiento, funciones y capacidad opcional para las necesidades de medición. Los modos de medición se combinan con el rendimiento del equipo para proporcionar soluciones personalizadas para su aplicación. El analizador de espectros se muestra en la Figura: C.9.

Figura C.9: Analizador de espectros: HP-8593E

- Realiza pruebas complejas de forma sencilla y rápida con personalidades de medición que admiten aplicaciones como televisión/transmisión por cable, prueba de componentes, compatibilidad electromagnética, ondas de luz y comunicaciones inalámbricas.
- Caja de tarjeta de cuatro ranuras.
- Pantalla dividida.
- Rutinas de medición con un solo botón.
- Funciones de medición avanzadas.
- Interfaces duales.

C.5.2. Tarjeta para mediciones VNA MegiQ VNA-Sandbox y VNA-0460e

Un VNA o analizador vectorial de redes permite el estudio de las características de señales eléctricas, enfocándose en los parámetros de dispersión y reflexión. Permite obtener mediciones de amplitud y fase. Para las pruebas de antenas y de amplificadores de bajo ruido, se utilizará el equipo VNA-0460e mostrado en la Figura C.10.

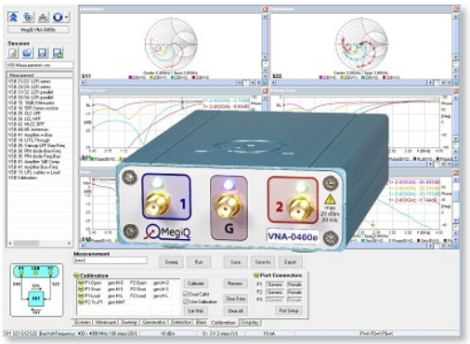


Figura C.10: VNA MegiQ VNA-Sandbox

El analizador vectorial de redes es un dispositivo que a diferencia de la mayoría de los multímetros y osciloscopios necesita de un uso y cuidado específico, caso contrario se podría dañar o dar lugar a resultados sin sentido.

Para empezar a utilizar el VNA se debe realizar la calibración del dispositivo por medio del kit de calibración incorporado. El kit cuenta con redes de 1 puerto, atenuadores de 2 puertos, filtros, un componente de amplificador activo y 2 antenas.

El VNA-0460e trabaja hasta una frecuencia de 6 GHz, es un analizador de red vectorial bidireccional de 2½puertos que se controla desde una PC a través de un conector USB. Este es ideal para adquirir mediciones de todo tipo como antenas, atenuadores, amplificadores, etc. Su rango de frecuencia incluye las bandas de telecomunicaciones más populares como GSM-GPRS-LTE, Wifi, DECT, GPS, ISM, Zigbee, Bluetooth.

C.6. NI-USB-5681

El sensor de potencia RF mostrado en la Figura C.11, trabaja en el rango desde los 10 MHz hasta los 18 GHz y en un rango de potencia desde los -40 dBm a +20 dBm. Este sensor permite realizar medidas muy precisas de distintas señales que van desde fuentes de un solo tono a múltiples tonos a formas de onda digitales y complejas de banda ancha.

Figura C.11: Sensor de Potencia NI USB-5681

Bibliografía

[1] I. . W. Group, *IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture*, 2014.

[2] M. Fogue, P. Garrido, F. J. Martinez, J. C. Cano, C. T. Calafate, P. Manzoni, y M. Sanchez, "Prototyping an automatic notification scheme for traffic accidents in vehicular networks," in *IFIP Wireless Days*, vol. 1, num. 1, 2011.

[3] J. A. Sánchez Sánchez, "Redes vehiculares aplicadas a la movilidad inteligente y sostenibilidad ambiental en entornos de ciudades inteligentes," Ph.D. dissertation, 2017.

[4] J. Gabarron, E. Lopez, y J. Haro, "Evaluacion de mecanismos de priorizacion en 802.11 p con vhdl," *VII Jornadad Ing. Telemat*, pp. 190–196.

[5] O. Orozco y G. Ramírez, "Aplicaciones para redes VANET Enfocadas en la Sostenibilidad Ambiental, una Revisión Sistemática," *Ciencia e Ingeniería Neogranadina*, vol. 24, num. 2, 2014.

[6] M. Gerla y L. Kleinrock, "Vehicular networks and the future of the mobile internet," *Computer Networks*, vol. 55, num. 2, pp. 457–469, feb 2011.

[7] V. Jindal, K. Mahavidyalaya, y P. Bedi, "Vehicular Ad-Hoc Networks: Introduction, Standards, Routing Protocols and Challenges," *International Journal of Computer Science Issues*, vol. 13, num. 2, pp. 44–55, 2016.

[8] D. F. Astudillo Salinas, "Téléchargement de Contenus dans les réseaux véhiculaires (Doctoral dissertation, École Doctorale Mathématiques, Informatique et Télécommunications (Toulouse)," *Sciences-New York*, vol. 5, pp. 0–180, 2013.

[9] Great Scott Gadgets, "Great Scott Gadgets - HackRF One," 2017. [En línea]. Disponible: https://greatscottgadgets.com/hackrf/

[10] ——, "ANT500-Great Scott Gadgets," 2016. [En línea]. Disponible: https://greatscottgadgets.com/ant500/

[11] GNU Radio Foundation, "GNU Radio - The Free & Open Source Radio Ecosystem ·GNU Radio," 2019. [En línea]. Disponible: https://www.gnuradio.org/

[12] A. M. S. Abdelgader y W. Lenan, "The physical layer of the IEEE 802. 11p WAVE communication standard: The specifications and challenges," *Lecture Notes in Engineering and Computer Science*, vol. 2, pp. 691–698, 2014.

[13] H. Zhou, *Wireless Ad-Hoc Networks*. InTech, dec 2012.

[14] M. C. Domingo Aladrén, *Diferenciación de servicios y mejora de la supervivencia en redes ad hoc conectadas a redes fijas*. Universitat Politècnica de Catalunya, 2005.

[15] J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the united states," *Proceedings of the IEEE*, vol. 99, num. 7, pp. 1162–1182, 2011.

[16] O. Olmedo, "Diseño de una red inalámbrica utilizando la tecnología wifi, para proveer servicio de internet a la parroquia Mindo," Ph.D. dissertation, Escuela politécnica nacional, 2010.

[17] R. Lopez y R. Guijarro, "Análisis del protocolo IEEE 802.11p en sistemas de comunicación vehiculares V2X," Ph.D. dissertation, Universitat Oberta de Catalunya, 2019.

[18] Y. Logro y M. Grace, "Simulación y análisis de los modelos de propagación para un canal implementado bajo el estándar IEEE 802.11P," Ph.D. dissertation, Escuela Politécnica Nacional, 2020.

[19] I. C. SOciety, "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments," *IEEE Std 802.11p-2010*, pp. 1–51, 2010.

[20] J. Siguenza, "Evaluación de la capa física del estándar IEEE 802.11p para redes vehiculares en un canal AWGN," Ph.D. dissertation, Escuela politécnica nacional, 2017.

[21] J. Casanova, "Evaluación de los protocolos IEEE 802.11p/1609.4 en entornos VANETs usando un modelo de movilidad realista," Ph.D. dissertation, Universidad Central Marta Abreu de las Villas, 2016.

[22] J. C. Lucero, "OFDM," 2017.

[23] C. Suárez, "Modulación multiportadora OFDM," *Ingeniería (Bogotá)*, vol. 6, num. 2, pp. 30–34, 2001.

[24] E. M. Yánez Mora, "Modulación OFDM y Sistemas Ópticos," 2016.

[25] E. Dahlman, S. Parkvall, y J. Sköld, *4G: LTE/LTE-Advanced for Mobile Broadband*, 2da ed., 2014.

[26] G. A. Álvarez, "Impacto de la distorsión armónica de amplificadores de potencia sobre la distorsión de señales moduladas digitalmente," 2014.

[27] S. G. Gago, "Metodos de modulacion digital," *Control*, num. 1, pp. 1–12, 2017.

[28] L. L. Hanzo, S. X. Ng, T. Keller, y W. Webb, *Quadrature Amplitude Modulation*, 2015.

[29] W. Sun, H. Zhang, C. Pan, y J. Yang, "Analytical study of the IEEE 802.11p EDCA mechanism," *IEEE Intelligent Vehicles Symposium, Proceedings*, num. Iv, pp. 1428–1433, 2013.

[30] G. Friedrich, G. Reggiani, R. Cayssials, S. Pellegrino, G. Velasquez, y L. Cofre, "Adaptación del Protocolo EDCA para Sistemas en Tiempo Real," *Argentine Symposium on Technology*, vol. 14, pp. 69–80, 2013.

[31] S. A. Malik, M. A. Shah, S. A. Khan, M. Jahanzeb, U. Farooq, y M. Adnan Khan, "Performance evaluation of IEEE 802.11p MAC protocol for VANETs," *Australian Journal of Basic and Applied Sciences*, vol. 4, num. 8, pp. 4089–4098, 2010.

[32] D. Jiang y L. Delgrossi, "IEEE 802.11p: Towards an international standard for wireless access in vehicular environments," *IEEE Vehicular Technology Conference*, pp. 2036–2040, 2008.

[33] J. S. Oñate Monta y D. P. Zambrano Herrera, "Diseño e implementación de voz sobre ip (voip) en redes inalámbricas wifi (wireless fidelity) bajo el estándar de la ieee 802.11 gy superiores." 2010.

[34] C. Avila, "Implementación de un Receptor OFDM IEEE 802 . 11 basado en SDR," Ph.D. dissertation, Universidad Nacional Autónoma de México, 2015.

[35] I. Analog Devices, "Software-Defined Radio Solutions from Analog Devices," p. 4, 2014.

[36] Ettus, "USRP B210 USB Software Defined Radio (SDR) - Ettus Research," 2019. [En línea]. Disponible: https://www.ettus.com/all-products/x310-kit/

[37] F. M. Paulo Delrivo, "Radio Definida por Software sobre BladeRF," 2010.

[38] "bladeRF 2.0 micro - Nuand." [En línea]. Disponible: https://www.nuand.com/bladerf-2-0-micro/

[39] H. Warehouse, "HackRF One Bundle - Hacker Warehouse." [En línea]. Disponible: https://hackerwarehouse.com/product/hackrf-one-kit/

[40] A. Store, "Arduino UNO Rev3," 2014. [En línea]. Disponible: https://store.arduino.cc/usa/arduino-uno-smd-rev3

[41] M. Electronic, "Tarjeta de Desarrollo Arduino Uno - R3." [En línea]. Disponible: https://www.mcielectronics.cl/shop/product/arduino-uno-r3-arduino-10230

[42] E. K. Arana Ortega y I. A. Portillo García, "Diseño de un arreglo fasorial de antenas de parche con técnica de reducción de estructura de microcinta imperfecta," jan 2015.

[43] C. A. Balanis, *Antenna theory: analysis and design*. John wiley & sons, 2016.

[44] L. Hu, K. Qian, y B. Wang, "Research of high efficiency x-band power amplifier," in *2015 8th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 1. IEEE, 2015, pp. 129–132.

[45] C. C. R. G. Mil-c, "Coaxial Cable RG178 MIL-C-17," p. 2015, 2015.

[46] "Embedded 5.8GHz with U.FL(IPEX) cable – Hz Labs." [En línea]. Disponible: https://hertzianlabs.com/products/embedded-5-8ghz-with-u-flipex-cable

[47] "LNA." [En línea]. Disponible: https://www.joom.com/en/products/5f378f97d784b201066af2da

[48] "PIGTAIL SMA, R-178." [En línea]. Disponible: https://www.zoominformatica.com/cable-pigtail-mmcx-a-rp-sma-hembra-invertido.html

[49] "Conectores SMA." [En línea]. Disponible: https://www.amazon.com/-/es/Conectores-Conector-Adaptador-coaxial-Extensi{ó}n/dp/B07FRGZVL9

[50] A. B. Sergienko, "Software-defined radio in MATLAB Simulink with RTL-SDR hardware," in *International Conference on Computer Technologies in Physical and Engineering Applications, ICCTPEA 2014 - Proceedings*, 2014, pp. 160–161.

[51] National Instruments, "LabVIEW NI." [En línea]. Disponible: https://www.ni.com/es-cr/shop/labview.html

[52] B. Ayyappan y P. Mohan Kumar, "Vehicular Ad Hoc Networks (VANET): Architectures, methodologies and design issues," in *2016 2nd International Conference on Science Technology Engineering and Management, ICONSTEM 2016*. IEEE, mar 2016, pp. 177–180.

[53] L. Guillermo, "Vehicle-to-Vehicle/Vehicle-to-Infrastructure Control," *The Impact of Control Tecnology*, 2011.

[54] C. N. Van Phu, N. Farhi, H. Haj-Salem, y J. P. Lebacque, "A vehicle-to-infrastructure communication based algorithm for urban traffic control," *5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2017 - Proceedings*, pp. 654–656, 2017.

[55] A. Jacobi, G.-W. Torng, J. L. Craig, Noblis, D. of Transportation, y F. T. Administration, "Transit Vehicle-to-Infrastructure (V2I) Assessment Study," num. July, p. 52p, 2015.

[56] P. Aucancela y E. Efrén, "Diseño de una red de comunicación vehicular inteligente, integrando la tecnología ad-hoc con LTE, para la movilidad en la zona urbana de la ciudad de Cuenca," Ph.D. dissertation, Universidad Politéctina Salesiana, 2015.

[57] E. Ruiz Alonso, "Diseño e Implementación del protocolo de notificación electrónica basado en redes VANET para E-SAVE," Universidad Carlos III de Madrid, Tech. Rep., 2013.

[58] J. Pereira, "Evaluation of IEEE 802.11a/g/p Transceiver for SDR," Ph.D. dissertation, Universidade do porto, 2015.

[59] B. Bloessl, M. Segata, C. Sommer, y F. Dressler, "Performance Assessment of IEEE 802.11p with an Open Source SDR-Based Prototype," *IEEE Transactions on Mobile Computing*, vol. 17, num. 5, pp. 1162–1175, 2018.

[60] J. R. Machado-Fernández, "Software Defined Radio: Basic Principles and Applications," *Revista Facultad De Ingeniería*, vol. 24, num. 38, p. 79, 2014.

[61] M. Mossman, "HackRF One · mossmann/hackrf Wiki · GitHub," 2017. [En línea]. Disponible: https://github.com/mossmann/hackrf/wiki/HackRF-One

[62] IBM/dsrc, "GitHub - IBM/dsrc: Dedicated short-range communication module for V2V using 802.11p." [En línea]. Disponible: https://github.com/IBM/dsrc

[63] C. org, "Index of files v3.7." [En línea]. Disponible: https://cmake.org/files/v3.7/?C=M;O=A

[64] M. M. Alam, M. R. Islam, M. Y. Arafat, y F. Ahmed, "FER performance evaluation and enhancement of IEEE 802.11 a/g/p WLAN over multipath fading channels in GNU radio and USRP N200 environment," *KSII Transactions on Internet and Information Systems*, vol. 12, pp. 178–203, 2018.

[65] B. Bloessl, M. Segata, C. Sommer, y F. Dressler, "Decoding IEEE 802.11a/g/p OFDM in software using GNU radio," *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, pp. 159–161, 2013.

[66] C. H. Liu, "On the Design of OFDM Signal Detection Algorithms for Hardware Implementation," *Conference Record / IEEE Global Telecommunications Conference*, vol. 2, pp. 596–599, 2003.

[67] T. Wang, A. Hussain, Y. Cao, y S. Gulomjon, "An improved channel estimation technique for ieee 802.11p standard in vehicular communications," *Sensors (Switzerland)*, vol. 19, num. 1, pp. 1–22, 2019.

[68] G. T. Grc y G. Radio, "Guided Tutorial GRC," 2016. [En línea]. Disponible: https://wiki.gnuradio.org/index.php/Guided_Tutorial_GRC

[69] GrC, "Embedded Python block example," 2016. [En línea]. Disponible: https://wiki.gnuradio.org/index.php/Embedded_Python_Block

[70] G. R. GrC, "Python: module," 2016. [En línea]. Disponible: https://wiki.gnuradio.org/index.php/Python_Module

[71] Á. Monteros, "Diseño y elaboración de prácticas de laboratorio para la manteria de Fundamentos de Comunicaciones usando radio definida por software," Ph.D. dissertation, Escuela Politécnica Nacional, 2019.

[72] J. Carpio López de Castro *y otros*, "Implementación de un sistema de comunicaciones basado en sdr mediante gnu radio," B.S. thesis, 2017.

[73] M. Mossman, "hackrf one pictures," 2017. [En línea]. Disponible: https://github.com/fd0/hackrf-one-pictures/blob/master/img_0009.jpg

[74] J. Rodríguez de Haro, "Análisis software y hardware del SDR HackRF One," p. 23, 2017.

[75] Tolocka, "Módulo de 4 relés para Arduino," 2017. [En línea]. Disponible: https://www.profetolocka.com.ar/2015/05/09/modulo-de-4-reles-para-arduino/