



**Universidad  
Politécnica  
de Cartagena**



**ETS  
Ingeniería de  
Telecomunicación**

Universidad Politécnica de Cartagena

---

# **ESTUDIO EXPERIMENTAL DE LA TÉCNICA DE FREQUENCY HOPPING EN REDES WIFI**

---

*Jose Cavas Alcaraz*

Dirigido por:

María Dolores Cano Baños  
José Luis Gómez Tornero

Colaborador:

Juan Carlos Jacobo Aarnoutse Sánchez

27 de abril de 2020



*A mis padres, por hacer de mi lucha, la suya propia.*

*“A veces las historias no van del descubrimiento de nuevos mundos,  
sino de lo que hay en lugares que creías vacíos”*

# Índice general

Resumen	2
Índice de ilustraciones	3
Índice de tablas	4
Capítulo 1: Introducción.	5
1.1. Redes inalámbricas.	5
1.2. Conceptos generales.	6
1.3. Objetivo del proyecto.	6
1.4. Contenido del proyecto.	7
Capítulo 2: Tecnologías.	8
2.1. Wi-Fi.	8
2.2. Frequency Hopping Spread Spectrum (FHSS).	10
2.3. Beamforming.	11
2.4. Lenguajes de programación.	12
Capítulo 3: Configuración y uso de Frequency Hopping en interfaces Wi-fi.	14
3.1. Pruebas experimentales: primer escenario.	14
3.1.1. Descripción de componentes.	14
3.1.2. Resultados.	21
3.2. Pruebas experimentales: segundo escenario.	22
3.2.1. Descripción de componentes.	23
3.2.2. Resultados.	26
Capítulo 4: Automatización del proceso en diferentes escenarios.	28
4.1. Estudio de parámetros de QoS.	28
4.2. Desarrollo de automatización.	29
Capítulo 5: Conclusión y líneas futuras.	31
Bibliografía.	33
Anexo.	34

## Resumen

En este proyecto se van a estudiar las capacidades y propiedades de las redes inalámbricas, más concretamente en el protocolo de comunicaciones Wi-fi. También pondremos a prueba el método de Frequency Hopping Spread Spectrum y comprobaremos lo eficaz que puede llegar a ser para el desarrollo de aplicaciones tales como la localización indoor o las flying networks.

Para ello, utilizaremos un tipo de antena especial llamado Leaky Waves, que nos ayudará a aplicar mejor el método de beamforming para las conexiones inalámbricas. Con estas herramientas se propone comprobar, mediante pruebas empíricas, que el uso del Frequency Hopping en estas antenas se comporta con naturaleza similar entre los mundos digital y analógico.

Una vez se verifique esta posibilidad, el siguiente paso será el de automatizar, mediante el conocimiento de parámetros de calidad de servicio, el cambio de canal en una comunicación para favorecerla o, en otros casos, para probar el medio en busca de valores que nos ayuden a localizar un dispositivo.

## Abstract

This Project will study the capabilities and properties of Wireless networks, more specifically in Wi-Fi communications. We will test the Frequency Hopping Spread Spectrum method and see how effective it can be for developing apps such as indoor localization or flying networks.

For that purpose, we will use a special type of antenna called Leaky Waves which will help us to do a better enforcement of the beamforming method for Wireless connections. With these tools, it is proposed to verify, through empirical tests, that the use of Frequency Hopping in these antennas behaves in a similar way between the digital and analog worlds.

Once this possibility is verified, next step will be to automate, through the knowledge of quality of service parameters, the hop of channel in a communication to promote it or, in other cases, to test the means of telecommunications in search of values that help us to locate a device.

# Índice de ilustraciones

Ilustración 1. Modo de funcionamiento Ad-hoc.....	5
Ilustración 2. Modo de funcionamiento en infraestructura.....	5
Ilustración 3. Características de las distintas tecnologías de localización indoor.....	6
Ilustración 4. Canales WiFi.....	8
Ilustración 5. Secuencia de salto de canal en Frequency Hopping.....	11
Ilustración 6. Comparación de radiación de una antena convencional respecto a una beamforming.....	11
Ilustración 7. Aplicación real de la técnica de beamforming.....	12
Ilustración 8. Icono lenguaje GNU Bash.....	12
Ilustración 9. Icono la plataforma MATLAB.....	13
Ilustración 10. Escenario de laboratorio 1.....	14
Ilustración 11. Cable coaxial SMA macho-hembra.....	15
Ilustración 12. Cable RJ-45.....	15
Ilustración 13. Atenuador.....	15
Ilustración 14. Conector en "T".....	15
Ilustración 15. Raspberry Pi 3 Model B.....	16
Ilustración 16. Adaptador USB inalámbrico TP-Link, modelo TL-WN722N.....	17
Ilustración 17. Portátil Lenovo Z50-70.....	18
Ilustración 18. Chipset AR9380 sin protección.....	19
Ilustración 19. Chipset AR9380 con protección.....	19
Ilustración 20. Wireless embedded board WPJ531.....	20
Ilustración 21. Medidas de RSSI puerto1.....	21
Ilustración 22. Medidas de RSSI puerto2.....	22
Ilustración 23. Escenario de laboratorio 2.....	23
Ilustración 24. Cámara anecoica.....	24
Ilustración 25. Antena Leaky Wave.....	24
Ilustración 26. Radiación de antena Leaky Wave.....	25
Ilustración 27. Mesa posicionadora Compact Table CT0800.....	25
Ilustración 28. Diagrama de radiación digital de la antena.....	27
Ilustración 29. Comparativa diagramas de radiación analógico y digital.....	27
Ilustración 30. Ejecución de comando iw station.....	28
Ilustración 31. Ecuación de reasociación para niveles de throughput.....	29
Ilustración 32. Throughput vs potencia recibida [21].....	29
Ilustración 33. Índice de MCS para el estándar 802.11ac [22].....	30
Ilustración 34. Flujograma de estados del programa.....	30

## Índice de tablas

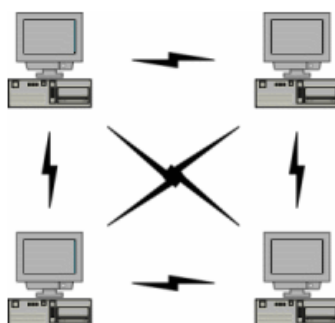
Tabla 1. Estándares 802.11.....	10
Tabla 2. Especificaciones Raspberry Pi 3 Model B.....	16
Tabla 3. Especificaciones amplificador USB TP-Link TL-WN722N. ....	17
Tabla 4. Característica y especificaciones portátil Lenovo Z50-70.....	19
Tabla 5. Característica y especificaciones del chipset inalámbrico AR9380. ....	20
Tabla 6. Especificaciones de Wireless embedded board WPJ531.....	21
Tabla 7. Características de la mesa posicionadora Compact Table CT0800.....	25

# Capítulo 1: Introducción.

## 1.1. Redes inalámbricas

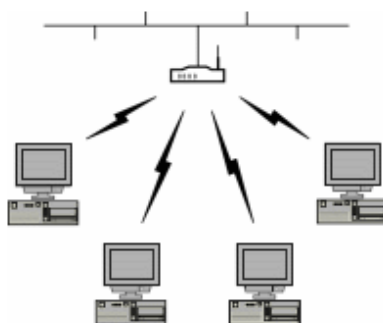
Las redes inalámbricas son aquellas que se comunican por medios no guiados a través de ondas electromagnéticas. Esta comunicación se efectúa mediante antenas que actúan, en su forma más simple, como receptor y emisor de una transmisión. El empleo de estas redes no sólo se basa en las comunicaciones de datos, también pueden utilizarse para domótica, sensorización y, como en el caso de este TFG, para la localización [1].

En las redes inalámbricas existen dos modos de funcionamiento: ad-hoc e infraestructura. En el funcionamiento ad-hoc de una red actúa con conexiones punto a punto entre distintos dispositivos con capacidades inalámbricas, sin la existencia o necesidad de puntos de acceso.



*Ilustración 1. Modo de funcionamiento Ad-hoc*

El modo infraestructura se gestiona a través de un punto de acceso y el resto de los dispositivos contactan con él para que los enrute hacia el camino correcto. En este caso tenemos la ventaja de poder estructurar estratégicamente la red, de forma que podamos tener mayor rango, pudiendo incluso añadir a la red más puntos de acceso que hagan de repetidores de la señal.



*Ilustración 2. Modo de funcionamiento en infraestructura*

Va a ser en las redes inalámbricas y en su funcionamiento donde nosotros vamos a apoyarnos para el estudio de técnicas de comunicaciones como es el frequency hopping. De aquí partiremos para comprobar que se puede asemejar el funcionamiento analógico de un tipo de antena, Leaky Wave, al funcionamiento digital del protocolo de comunicaciones Wi-fi, para después dar un objetivo a su comportamiento característico.



## 1.2. Conceptos generales

La técnica de Frequency Hopping es un método utilizado para cambiar de rango de frecuencias (canal) con el objetivo de mejorar la comunicación o impedir la pérdida de la misma. Este cambio debe producirse de forma síncrona entre emisor y receptor, siguiendo lo que se denomina patrón de salto.

La localización indoor trata de localizar de forma inalámbrica dispositivos o personas en un espacio dado. Existen distintas tecnologías para este uso, como son las etiquetas RFID o las aplicaciones wifi y bluetooth. Algunos usos prácticos son: seguimiento del comportamiento de consumidores en supermercados, tráfico ferroviario u ocupación en edificios de oficinas. Nosotros vamos a utilizar la tecnología Wi-Fi para la localización de dispositivos y, además, utilizaremos un tipo de antena especial llamada Leaky Waves. Con esta antena podemos aplicar de una forma más eficiente al técnica de frequency hopping anteriormente descrita.












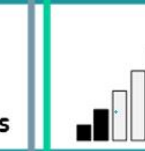




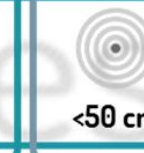
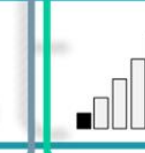
	Indoor/ Outdoor	Alcance / Cobertura	Consumo / Autonomía	Precisión Indoor	Coste localizador
	 Con infraestructura	 <150 Metros	 -	 5-15 Metros	
	 Con infraestructura	 <30 Metros	 Autónomo 2-4 Años	 1-3 Metros	
	 Sólo en pórticos	 <8 Metros	 Autónomo >4 Años	 <50 cm	

Ilustración 3. Características de las distintas tecnologías de localización indoor.

Las antenas Leaky Waves, también llamadas antenas permeables o de onda de fuga, utilizan una estructura que soporta la propagación de la onda a lo largo de su forma, es decir, radiando o “goteando” a lo largo de su estructura. Este efecto se llama fuga de onda, se produce cuando gran parte de la onda electromagnética generada se “fuga” o sale de la superficie de la antena.

## 1.3. Objetivo del proyecto

El objetivo de este proyecto se basa en el estudio de la técnica de Frequency Hopping (FH), el cual, como se mencionó anteriormente, trata de mantener una conexión entre un emisor y un receptor cambiando entre los distintos canales de forma síncrona, permitiendo que la comunicación continúe sin cortes.

Mediante la aplicación de unos escenarios de pruebas, se va a comprobar empíricamente el funcionamiento de la técnica anteriormente mencionada en conjunto con el uso de una antena Leaky Wave, esto, añadido al estudio de parámetros de QoS (Quality of Service) o calidad de

servicio, nos va a permitir la creación de aplicaciones con posibles orientaciones para la localización de dispositivos e incluso para favorecer comunicaciones Wi-Fi.

## **1.4. Contenido del proyecto**

Como hemos podido ver en el capítulo 1 se hace una introducción general, tanto de las tecnologías utilizadas en el proyecto como de los objetivos. Ahora veremos el contenido de cada capítulo.

En el capítulo 2 vamos a ver en profundidad qué tecnologías se van a utilizar en este TFG para poder contextualizar desde el inicio los principios que debemos entender. Es en el capítulo 3 donde se van a describir el proceso de cada prueba que se hace con la instrumentación de laboratorio, las pruebas realizadas y los resultados obtenidos. Cuando estas pruebas hayan acabado podremos pasar al capítulo 4 para ver qué aplicaciones se le puede dar a las características que se esperaban corroborar en el anterior capítulo, realizando una automatización de los procesos de frequency hopping en base a determinados parámetros de la calidad de servicio de la comunicación. Por último, en el último capítulo, se hace una síntesis de todo lo realizado, analizando qué se ha conseguido y qué se espera conseguir en posibles líneas futuras.

# Capítulo 2: Tecnologías.

## 2.1. Wi-Fi

El WiFi, Wireless Fidelity, se ha convertido en el estándar de redes inalámbricas dominante[2]. Esta tecnología dispone de ventajas que derivan en su mayoría de dicha popularidad, entre ellas encontramos la facilidad para conseguir productos que respondan a este estándar, lo que significa unos precios más permisivos. Una ventaja destacable es la movilidad a la hora de usar WiFi, la cual nos permite cambiar de punto de acceso sin la restricción de pérdida de conexión. Su propiedad inalámbrica también nos proporciona facilidad para proporcionar escalabilidad a una red; en el caso de una empresa, la entrada de nuevos miembros o el despliegue de un departamento no nos obliga, salvo restricción obligatoria, a cambiar el equipamiento ya desplegado. Por el contrario, para las redes por cable siempre habría que añadir mínimo una conexión física nueva.

Las principales desventajas de WiFi son la seguridad, el rango y la fiabilidad. Al ser una red inalámbrica, los datos que se intercambian viajan a través de ondas electromagnéticas utilizando el aire como medio, al contrario que con conexiones cableadas, este medio se considera inseguro en cuanto a la facilidad de acceso a él. Debido a esto, las redes WiFi deben estar provistas de otras herramientas para proporcionar seguridad a las comunicaciones, tales como la limitación de acceso a la red o el encriptado de la información intercambiada. El rango de alcance de estas redes es limitado, necesitando otros dispositivos que aumenten el radio de la red para poder cubrir zonas grandes como empresas o espacios públicos. La última desventaja es la de la fiabilidad, como se mencionó anteriormente, las conexiones WiFi utilizan ondas electromagnéticas para transmitir, esto hace que las comunicaciones queden sujetas a gran variedad de interferencias y a características negativas de la propagación de la onda, tales como la dificultad de atravesar objetos o el jamming [3].

El estándar IEEE 802.11 define la normativa de las redes inalámbricas y especifica principalmente dos rangos en los que se puede emitir, éstos son la banda de 2,4 GHz y la de 5 GHz. Nosotros vamos a trabajar con los 11 canales europeos de los 14 que se pueden utilizar en otras regiones; en la banda de 2,4 GHz podemos observar un rango desde 2,412 GHz hasta 2,462 GHz[4].

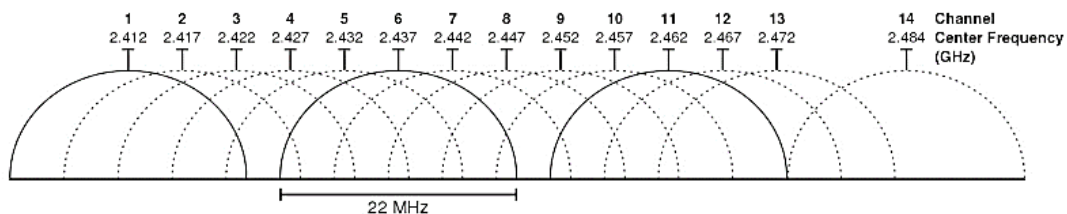


Ilustración 4. Canales WiFi

Actualmente, los estándares más utilizados son:

Estándares actuales Wi-Fi	
<ul style="list-style-type: none"> <li>● 802.11: Velocidad (teórica) = 2 Mbps Velocidad (práctica) = 1 Mbps Frecuencia = 2,4 GHz Ancho de banda = 22 MHz Alcance = 330 m</li> </ul>	<ul style="list-style-type: none"> <li>● 802.11a: Velocidad (teórica) = 54 Mbps Velocidad (práctica) = 22 Mbps Frecuencia = 5,4 GHz Ancho de banda = 20 MHz Alcance = 390 m</li> </ul>
<ul style="list-style-type: none"> <li>● 802.11b: Velocidad (teórica) = 11 Mbps Velocidad (práctica) = 6 Mbps Frecuencia = 2,4 GHz Ancho de banda = 22 MHz Alcance = 460 m</li> </ul>	<ul style="list-style-type: none"> <li>● 802.11g: Velocidad (teórica) = 54 Mbps Velocidad (práctica) = 22 Mbps Frecuencia = 2,4 GHz Ancho de banda = 20 MHz Alcance = 460 m</li> </ul>
<ul style="list-style-type: none"> <li>● 802.11n: Velocidad (teórica) = 600 Mbps Velocidad (práctica) = 100 Mbps Frecuencia = 2,4 GHz y 5,4 GHz Ancho de banda = 20/40 MHz Alcance = 820 m</li> </ul>	<ul style="list-style-type: none"> <li>● 802.11ac: Velocidad (teórica) = 6,93 Gbps Velocidad (práctica) = 100 Mbps Frecuencia = 5,4 GHz Ancho de banda = 80/160 MHz</li> </ul>
<ul style="list-style-type: none"> <li>● 802.11ad: Velocidad (teórica) = 7,13 Gbps Velocidad (práctica) = 6 Gbps Frecuencia = 60 GHz Ancho de banda = 2 MHz Alcance = 300 m</li> </ul>	<ul style="list-style-type: none"> <li>● 802.11ah: Velocidad (teórica) = 0,9 Gbps Ancho de banda = 2 MHz Alcance = 1000 m</li> </ul>
<ul style="list-style-type: none"> <li>● 802.11: Velocidad (teórica) = 2 Mbps Velocidad (práctica) = 1 Mbps Frecuencia = 2,4 GHz Ancho de banda = 22 MHz Alcance = 330 m</li> </ul>	<ul style="list-style-type: none"> <li>● 802.11a: Velocidad (teórica) = 54 Mbps Velocidad (práctica) = 22 Mbps Frecuencia = 5,4 GHz Ancho de banda = 20 MHz Alcance = 390 m</li> </ul>
<ul style="list-style-type: none"> <li>● 802.11b: Velocidad (teórica) = 11 Mbps Velocidad (práctica) = 6 Mbps Frecuencia = 2,4 GHz Ancho de banda = 22 MHz Alcance = 460 m</li> </ul>	<ul style="list-style-type: none"> <li>● 802.11g: Velocidad (teórica) = 54 Mbps Velocidad (práctica) = 22 Mbps Frecuencia = 2,4 GHz Ancho de banda = 20 MHz Alcance = 460 m</li> </ul>
<ul style="list-style-type: none"> <li>● 802.11n: Velocidad (teórica) = 600 Mbps Velocidad (práctica) = 100 Mbps</li> </ul>	<ul style="list-style-type: none"> <li>● 802.11ac: Velocidad (teórica) = 6,93 Gbps Velocidad (práctica) = 100 Mbps</li> </ul>

Frecuencia = 2,4 GHz y 5,4 GHz Ancho de banda = 20/40 MHz Alcance = 820 m	Frecuencia = 5,4 GHz Ancho de banda = 80/160 MHz
<ul style="list-style-type: none"> <li>● 802.11ad: Velocidad (teórica) = 7,13 Gbps Velocidad (práctica) = 6 Gbps Frecuencia = 60 GHz Ancho de banda = 2 MHz Alcance = 300 m</li> </ul>	<ul style="list-style-type: none"> <li>● 802.11ah: Velocidad (teórica) = 0,9 Gbps Ancho de banda = 2 MHz Alcance = 1000 m</li> </ul>

Tabla 1. Estándares 802.11.

## 2.2. Frequency Hopping Spread Spectrum (FHSS).

Para poder entender correctamente la técnica de Frequency Hopping (salto en frecuencia), también llamado Channel Hopping, debemos comprender la técnica de Spread Spectrum (espectro ensanchado).

El espectro ensanchado es una técnica de modulación digital que extiende una señal a través de múltiples frecuencias para facilitar la detección de la misma y evitar interferencias. De esta forma, podemos utilizar un ancho de banda de datos mayor con la misma potencia de señal [5], [6].

Frequency Hopping Spread Spectrum (salto en frecuencia con espectro ensanchado) es una tecnología de transmisión utilizada en redes inalámbricas que utiliza la técnica de espectro ensanchado haciendo saltar a la portadora en frecuencia [7]. En este método la señal de datos se modula con una señal portadora de banda estrecha, la cual salta en una secuencia pseudoaleatoria de frecuencia en frecuencia. Ambos puntos finales de la comunicación deben estar sincronizados.

El uso de esta tecnología de salto en frecuencia es una solución altamente válida para evitar interferencias o distorsión. También aumenta la capacidad de la señal, mejora la relación señal a ruido (SNR) y la eficiencia del ancho de banda de la comunicación. Gracias a que la comunicación salta en frecuencia es difícil interceptar la señal, aumentando así la privacidad de la transmisión. Podemos encontrar una modalidad de Frequency Hopping llamada Adaptive Frequency Hopping (APH), en este modo los dispositivos cambian de canal si éste se encuentra saturado para evitar así interferencias con otras comunicaciones.

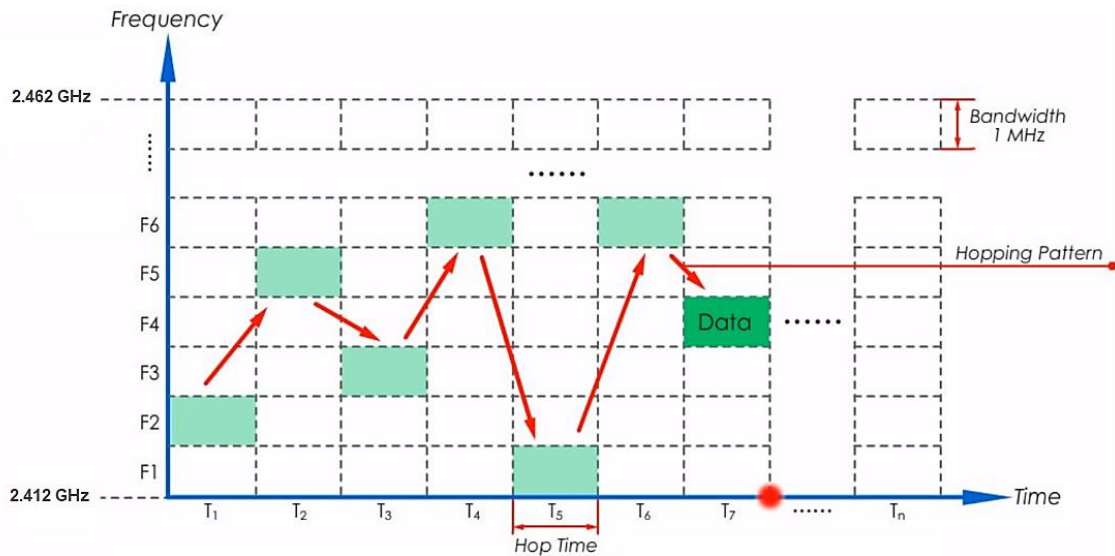


Ilustración 5. Secuencia de salto de canal en Frequency Hopping

En la *ilustración 7* podemos ver una secuencia de salto de una comunicación determinada a través del tiempo, donde observamos que el tiempo que reside la transmisión en un canal concreto se llama tiempo de salto y al camino de frecuencias que sigue una comunicación se llama patrón de salto.

### 2.3. Beamforming.

Esta técnica viene precedida del uso de antenas omnidireccionales. Estas antenas envían la energía en todas direcciones, a pesar de sus ventajas, las antenas omnidireccionales tienen como inconveniente principal que un solo canal está utilizando todo el espacio en la comunicación [8].

Para solucionar el problema comentado y capacitar de nuevas ventajas a la comunicación nace una nueva idea, enfocar la energía hacia el receptor, beamforming o formación de haz [9]. Esta técnica mejora el rendimiento de las redes inalámbricas a media distancia, de forma que se aprovechan de forma más efectiva los canales que favorecen más la comunicación que se está dando en ese momento.

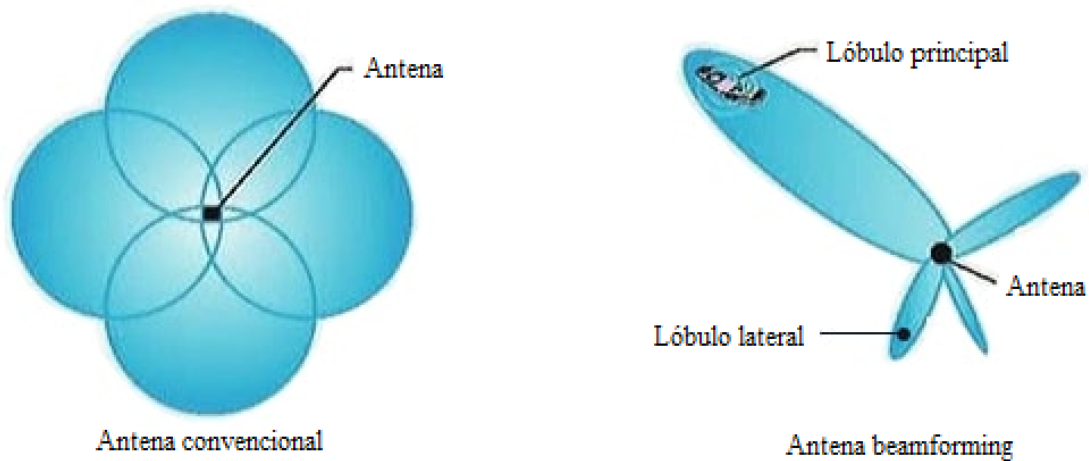


Ilustración 6. Comparación de radiación de una antena convencional respecto a una beamforming.

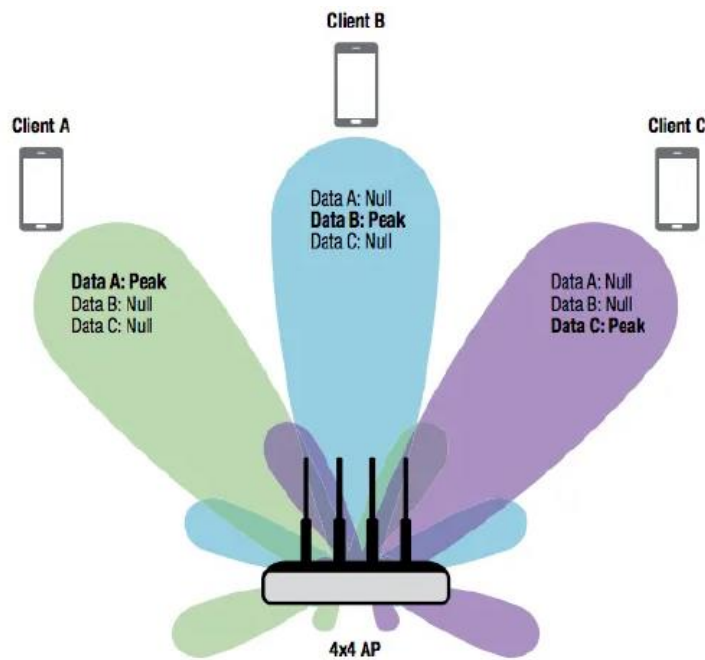


Ilustración 7. Aplicación real de la técnica de beamforming.

Esta tecnología se incorpora en el protocolo 802.11ac por primera vez y permite que el router emita la señal en la dirección en la que se encuentra un dispositivo. El router es capaz de enviar las ondas por múltiples caminos, técnica de multipath, y analizar así dónde se encuentran los dispositivos y qué parámetros de comunicaciones se reciben. Estos parámetros pueden ser la relación señal a ruido, interferencias de la comunicación u otros parámetros sobre el estado de la comunicación.

La forma en la que nuestro router se va a comunicar con el resto de los dispositivos cliente es usando la tecnología MU-MIMO. En este caso, el router establece comunicación con todos los dispositivos a la vez. Dentro de la especificación 802.11ac constan dos versiones, Wave 1 y Wave 2:

802.211ac Wave 1	802.211ac Wave 2
Banda de 5 GHz	Banda de 5 GHz
Canales: 20, 40 y 80 Mhz	Canales: 20, 40, 80, 80, 80-80 (160) MHz
Soporta hasta 4 streams de 450 Mbps	Soporta hasta 8 streams de 450 Mbps
256 QAM	256 QAM

## 2.4. Lenguajes de programación.

### GNU Bash

Para el control y gestión, así como el envío de datos para su posterior procesamiento de nuestro router se ha utilizado el intérprete de Bash, se trata de un lenguaje de comandos perteneciente al Shell de Linux. Este lenguaje se puede ejecutar desde una ventana de comandos o se puede leer una ristra de comandos Bash desde un archivo



Ilustración 8. Icono lenguaje GNU Bash.

de formato “.sh” [12]. El intérprete de Bash permite el uso de variables y estructuras de control, tales como condicionales o bucles, y funciones.

Las ventajas que ofrece Bash son las siguientes:

- Edición de línea de comando.
- Historial de comandos de tamaño ilimitado.
- Control de trabajo.
- Funciones de Shell y Alias.
- Matrices indexadas de tamaño ilimitado.
- Aritmética de enteros en cualquier base de dos a sesenta y cuatro.

### **MATLAB (Matrix Laboratory)**

Se trata de una plataforma orientada a la resolución de cálculos científicos y de ingeniería. Su lenguaje de programación, llamado M, está basado en matrices [13].

Permite el manejo y cálculo de matrices, la representación de datos mediante distintos tipos de gráficas, la comunicación entre otros programas o dispositivos (como en nuestro caso en este proyecto) y la creación de interfaces de usuario. Además, cuenta con otras herramientas que amplían la funcionalidad de esta plataforma como son Simulink y GUIDE.

Sus características principales son:

- Lenguaje de alto nivel para cálculos científicos y de ingeniería.
- Entorno de escritorio optimizado para la exploración iterativa, el diseño y la solución de problemas.
- Gráficas para visualizar datos y herramientas para crear diagramas personalizados.
- Aplicaciones para ajustar curvas, clasificar datos, analizar señales, ajustar sistemas de control y muchas otras tareas.
- Toolboxes complementarias para una amplia variedad de aplicaciones científicas y de ingeniería.
- Herramientas para crear aplicaciones con interfaces de usuario personalizadas.
- Interfaces para C/C++, Java®, .NET, Python, SQL, Hadoop y Microsoft® Excel®.
- Opciones de implementación open-source para compartir programas de MATLAB con los usuarios finales.



*Ilustración 9. Icono de la plataforma MATLAB.*



## Capítulo 3: Configuración y uso de frequency hopping con interfaces WiFi.

En este proyecto hemos preparado dos escenarios, explicados en profundidad más adelante en esta memoria, para estudiar el funcionamiento tanto de la antena que utilizamos, diseñada en la Universidad Politécnica de Cartagena, como de la técnica de Frequency Hopping aplicada a ella.

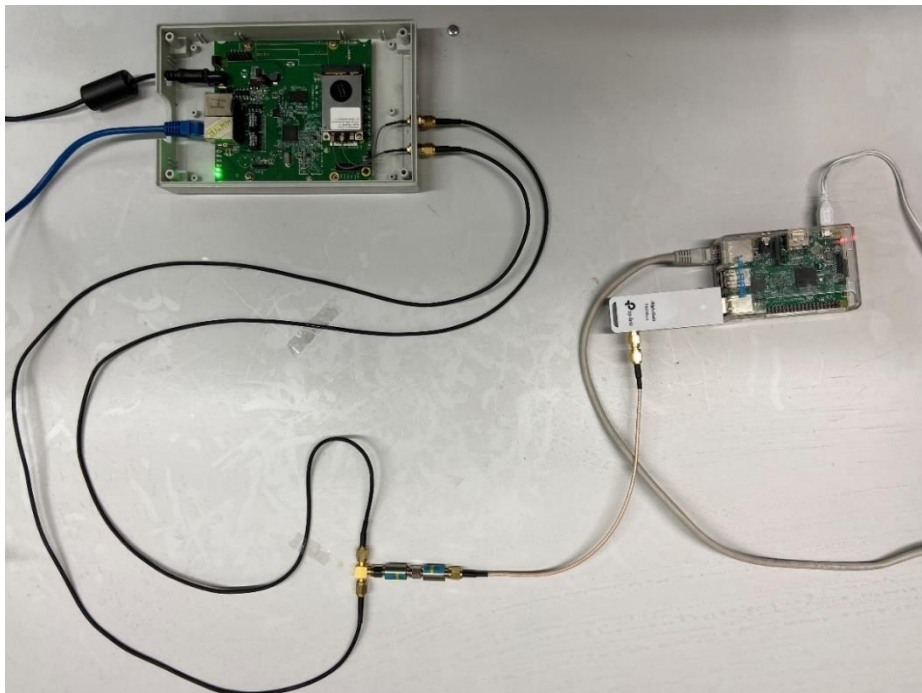
El primer escenario se va a utilizar para la comprobación del correcto funcionamiento de los componentes, verificando que la potencia con la que enviamos corresponde al valor de RSSI obtenido. Tanto en este primer escenario como en el segundo, la recepción de los datos ocurre a través de una conexión UDP desde el chipset o router hasta un ordenador portátil conectado a él.

Nuestro segundo escenario ocurre en la cámara anecoica, vamos a ubicar nuestra antena objetivo de estudio en una mesa posicionadora y la enfrentaremos hacia nuestro receptor. En este contexto, la mesa gira hacia ángulos fijados previamente mientras la antena emite y almacenaremos los valores RSSI obtenidos.

### 3.1. Pruebas experimentales: primer escenario.

#### 3.1.1. Descripción de componentes.

En este apartado vamos a realizar una descripción de los distintos dispositivos y componentes que se utilizan para la realización del escenario 1.



*Ilustración 10. Escenario de laboratorio 1.*

## Cableado y conexiones

Para el cableado se han utilizado tres tipos distintos: cable coaxial, cable ethernet RJ-45 y los correspondientes a la alimentación de dispositivos.



Ilustración 11. Cable coaxial SMA macho-hembra.



Ilustración 12. Cable RJ-45.

- Cable coaxial: se trata de un cable utilizado para transmitir señales de alta frecuencia. Tiene dos conductores concéntricos, uno central llamado vivo, y un conductor externo o malla que sirve de referencia a tierra y retorno de corrientes. Todo ello se encuentra envuelto por una cubierta exterior de plástico o goma. Puede alcanzar velocidades de hasta 2 Gbps y sus aplicaciones más comunes son el transporte de señal de televisión y su uso en la red urbana [10].
- Cable RJ-45: también llamado par trenzado, consiste en ocho hilos trenzados dos a dos según la función de cada par, es decir, si se destinan a transmisión, recepción u otras funcionalidades. Se pueden distinguir dos tipos según la protección, sin apantallar (UTP) o apantallado (STP), siendo este último el más robusto con sus pares protegidos de forma independiente. Puede alcanzar una velocidad de 1 Gbps [11].
- Los cables de alimentación no se van a definir, son los específicos de los dispositivos que se exponen posteriormente.

Para conectar las salidas de los dos puertos de nuestro router vamos a utilizar un conector en "T". A su salida se va a conectar varios atenuadores para simular un medio coherente con pérdidas, uno de ellos con una atenuación de 20 dB y otro, 10 dB.



Ilustración 13. Atenuador.



Ilustración 14. Conector en "T".

## Raspberry Pi 3 Model B+

En este escenario tendremos como receptor una Raspberry Pi 3 Model B+. Este tipo de ordenadores simples se utilizan para todo tipo de proyectos gracias a su versatilidad y tamaño. Se compone de un procesador 1,4 GHz quad-core ARM Cortex A53 de 64 bit, 1 GB de memoria RAM soldada, un puerto HDMI, cuatro puertos USB 2.0, ranura para tarjeta MicroSD y conectores CSI y DSI.

Las características que más nos interesan son las inalámbricas, cuenta con una tarjeta que permite el uso de Wi-Fi 802.11ac de doble banda y Bluetooth 4.2. También cuenta con un puerto ethernet de 300 Mbps.

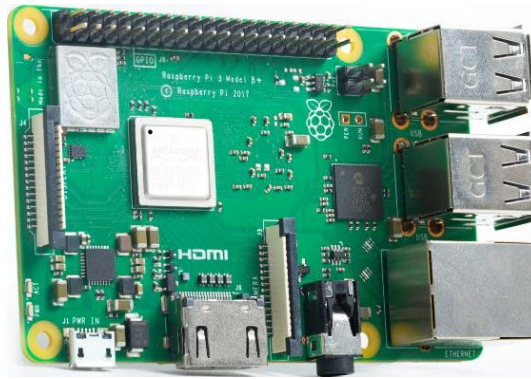


Ilustración 15. Raspberry Pi 3 Model B.

En nuestro caso contamos con una tarjeta SD como almacenamiento con una capacidad de 16 GB para guardar los archivos que generaremos en las pruebas de laboratorio. El sistema operativo que utiliza nuestra Raspberry es una distribución de Linux basada en Debian llamada Raspbian. Para que sea un receptor apropiado se ha desactivado el control de potencia automático, de forma que no intente compensar la comunicación cuando reciba una señal débil.

	Raspberry Pi 3 Model 3
Procesador	Broadcom BCM2873
CPU Core	Quadcore ARM Cortex-A53, 64 bits
RAM	1 GB LPDDR2 SDRAM
GPU	400 MHz VideoCore IV®
Velocidad de reloj	1,4 GHz
Conectividad de red	1 x 10 / 100 Gigabit Ethernet (RJ45 Port)
Conectividad inalámbrica	802.11.b/g/n/ac Wireless LAN (WiFi) y Bluetooth 4.2
Puertos USB	4 x USB 2.0 Ports
GPIOs	2 x 20 Pin Header
Interfaz de monitor	DSI 15 Pin / HDMI salida / Composite RCA
Interfaz de cámara	15-pin MIPI
Almacenamiento	MicroSD
Fuente de alimentación	2,5 A @ 5 V

Tabla 2. Especificaciones Raspberry Pi 3 Model B

## Adaptador USB inalámbrico

Para las pruebas vamos a desactivar la tarjeta inalámbrica propia de la Raspberry y la vamos a sustituir por un adaptador USB inalámbrico de alta ganancia de la marca TP-Link, modelo TL-WN722N.

El uso de este dispositivo se debe a la necesidad de utilizar una antena de polarización vertical como receptor, al igual que la antena emisora; si no fuese así, las antenas no podrían comunicarse [11]. Para entender esto debemos profundizar en la radiación de las antenas. El campo eléctrico de las ondas electromagnéticas que radian las antenas determina la polarización u orientación de las ondas de radio. La polarización electromagnética define el plano por el que propaga la onda, dejando así tres tipos de propagación: lineal, circular y elíptica.

En nuestro caso, el tipo de polarización que nos ocupa es la lineal. Cuando el vector eléctrico, el cuál es componente del campo electromagnético, oscila en una dirección que forma un ángulo constante decimos que las ondas están linealmente polarizadas. Si la polarización se produce en un campo perpendicular al horizonte se trata de la polarización vertical.



Ilustración 16. Adaptador USB inalámbrico TP-Link, modelo TL-WN722N.

Una vez sabemos que nuestra antena es de polarización vertical, debemos configurar su tarjeta de red para que pueda comunicarse con el emisor. Los equipos trabajan mediante el protocolo TCP/IP (Transmission Control Protocol/Internet Protocol), y para que nuestros equipos funcionen correctamente debemos configurar un mínimo de tres parámetros, los cuáles son la dirección IP, la máscara de subred y la puerta de enlace. La dirección IP es una dirección lógica de 32 bits que sirve para identificar de forma unívoca a un equipo en una red. A su vez, la máscara de subred sirve para que los equipos pertenecientes a una misma red se puedan comunicar compartiendo el mismo identificador de red, es decir, la red principal. Por último, la puerta de enlace es quien sirve de enlace entre dos redes, permitiendo a un host de una red acceder a otra con distinto identificador de red en caso necesario. Ahora que está comprendido el concepto, sabemos que ambos dispositivos deben configurarse con la misma red y máscara de subred (192.168.2.0/24), además este adaptador tiene como puerta de enlace la dirección del router emisor.

Características generales	
Rango de frecuencias	Rango de frecuencias
Tasas de señal inalámbrica	Tasas de señal inalámbrica
Potencia de transmisión inalámbrica	Potencia de transmisión inalámbrica
Modo de funcionamiento	Modo de funcionamiento
Antena	Antena

Tabla 3. Especificaciones amplificador USB TP-Link TL-WN722N.

## Ordenador portátil Lenovo Z50-70

Vamos a necesitar un ordenador portátil para varias tareas. Una de ellas será la de controlar, por medio de una conexión SSH (Secure Shell), el router que hace el papel de emisor. Otra función es la de recibir a través del protocolo UDP (User Datagram Protocol) la información que vamos a procesar para ver si efectivamente el escenario produce los resultados esperados.

El protocolo SSH nos permite acceder a un dispositivo a través de una conexión remota, de esta forma, se crearon los archivos ejecutables o scripts que realizan las acciones de envío, cambio de canal y medidas entre otras.

UDP se ha utilizado como protocolo simple de nivel de transporte, dado que proporciona una interfaz muy sencilla entre la capa de red y al de transporte, no afecta a la capacidad de cómputo de nuestro router y permite que los resultados no se vean afectados por otras tareas.



Ilustración 17. Portátil Lenovo Z50-70.

Características y especificaciones	
Procesador	Intel Core i7-4510 CPU @ 2.00 GHz 2.60 GHz
Arquitectura de bus	<ul style="list-style-type: none"> <li>• GPU: PCIe 2.0, 5GT / s</li> <li>• WLAN: PCIe 1.0, 2.5GT / s</li> <li>• RAM: DDR3L, 1600MT / s</li> <li>• DMI x 4, N / A</li> <li>• FDI x 2, N / A</li> <li>• LCD: eDP 1.2, 5.4GT / s</li> <li>• Puerto HDMI: HDMI 1.4, 1.65GT / s</li> <li>• HDD: SATA 3.0, 6 Gb / s</li> <li>• ODD: SATA 1.0, 1.5Gb / s</li> <li>• Puerto USB: USB 3.0, 5Gb / s</li> <li>• USB 2.0, 480Mb / s</li> <li>• BT: USB 2.0, 480Mb / s</li> <li>• Lector de tarjetas: SD3.0, 50Mb / s</li> <li>• VGA: DP a VGA</li> <li>• SMBus 2.0</li> <li>• LPC 1.1</li> <li>• HDA</li> <li>• SPI</li> </ul>
Chipset gráfico	Integrado / AMD Jel LE / AMD Jel Pro / nvidia N15S-GT / nvidia N15V-GM
Monitor	15.6 "16: 9, 1.366 x768 píxeles HD / 1.920 x 1.080 píxeles FHD
Memoria estándar	DDR3L, SODIMM x 1 / SODIMM x 2 (máximo 8 GB / 16 GB)
Disco duro	2.5 pulgadas, 7 mm / 9.5 mm SATA
	320 GB / 500 GB / 1TB 5.400 rpm
Puerto E/S	Conector de audio combinado x 1

	RJ45 x 1 Puerto HDMI x 1 Puerto USB 2.0 x 2, puerto USB 3.0 x 1 2 en 1 ranura x 1 (SD / MMC)
Ethernet	10/100 / 1,000 Mbps
Sistema operativo	Windows 10 x 64

Tabla 4. Característica y especificaciones portátil Lenovo Z50-70

## Chipset inalámbrico AR9380



Ilustración 19. Chipset AR9380 con protección.



Ilustración 18. Chipset AR9380 sin protección.

Se trata de una implementación inalámbrica desarrollada por Qualcomm que ofrece una solución LAN para el estándar 802.11n. Algunas de las características que reúne esta cuarta generación de soluciones Wi-Fi son:

- Ofrece un coste y forma que permiten impulsar el mercado de 802.11n.
- Configuraciones de banda única, banda doble y Multiple MIMO.
- Interfaces periféricas que pueden ser usadas en aplicaciones avanzadas de redes.
- Configuración de red sencilla.

Especificaciones AR9380	
Banda de frecuencias	2.4 / 5 GHz
Estándar de red	802.11a, 802.11b, 802.11g, 802.11n
Tecnología de modulación	OFDM con BPSK, QPSK, 16 QAM, 64 QAM; DBPSK, DQPSK, CCK
Codificación FEC	Convolution Code Low-Density Parity Check (LDPC)
Cifrado hardware	AES, TKIP, WEP
Calidad de servicio	802.11e
Interfaz de host	PCI-Express

Interfaz de periféricos	GPIO
Velocidades de datos soportadas	IEEE 802.11a 6 - 54 Mbps IEEE 802.11b 1 - 11 Mbps IEEE 802.11g 6 - 54 Mbps IEEE 802.11n 6.5 - 450 Mbps

Tabla 5. Característica y especificaciones del chipset inalámbrico AR9380.

## Wireless embedded board WPJ531

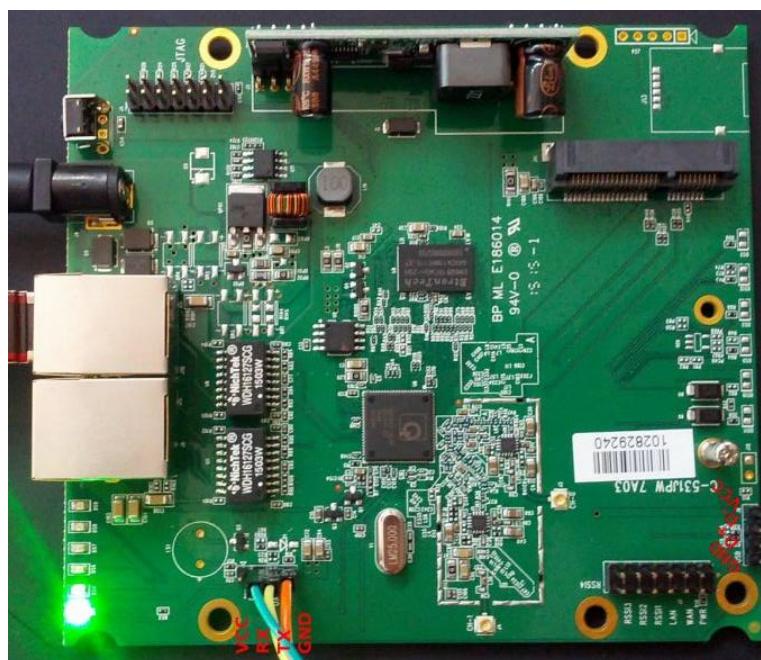


Ilustración 20. Wireless embedded board WPJ531.

La pieza fundamental de nuestro router sería la placa WPJ531. En ella se instalará el chipset AR9380 descrito arriba.

El sistema operativo que controla este dispositivo es OpenWRT. Se trata de un sistema operativo de Linux diseñado para dispositivos integrados, que permite la configuración personalizada del dispositivo para su uso en cualquier aplicación. Sobre este sistema operativo es donde programaremos con lenguaje Bash la rutina que cambiará de canal cada cierto número de muestras. Además, será el encargado de transmitir vía UDP todos los datos recogidos en la comunicación.

Especificaciones Wireless embedded board WPJ531	
Tipo de product	Wireless Embedded Board
Chipset original	QCA9531
Frecuencia CPU	650MHz
Arquitectura	MIPS 24Kc
Memoria del sistema	128MB DDR2
Memoria Flash	16MB
Estándar inalámbrico	802.11n, 802.11g, 802.11b
Canales MIMO	2x2

Rango de frecuencias	2.412 ~ 2.472 GHz
Potencia de salida	23dBm
Velocidad inalámbrica	300Mbps
Slot MiniPCI-e	v1.1
Interfaz	1x JTAG 14 Pin Connector, 1x MiniPCI-e Slot 9.2mm, 1x Serial Port 4 Pin Connector, 2x Fast Ethernet Port, 2x U.FL Connector
LED	14 PIN Connectors, 6x LED Indicators
Power over Ethernet	Passive PoE 24V (LV version) IEEE 802.3af/at or Passive PoE 24-56V (HV version)
Jack DC	12 – 24V
Botón de reset	Yes
Sistema operativo	CompexWRT, OpenWRT
Certificación	CE, FCC, REACH&RoHS Compliance
Consumo de potencia	6.9 Watts (Max)
Extra	Surge Arrestor, Watchdog Timer
Rango de temperatura	Operating: -20°C to 70°C, Storage: -40°C to 90°C
Humedad	Operating: 5% to 95% (non-condensing)
Dimensiones	117 x 105 x 17mm (W x H x D)

Tabla 6. Especificaciones de Wireless embedded board WPJ531

### 3.1.2.Resultados.

Se recogieron 400 valores de RSSI (Received Signal Strength Indicator), pero, dado que el chipset inalámbrico AR9380 tiene dos puertos, como se puede ver en la *Ilustración 9*, son 200 valores por puerto.

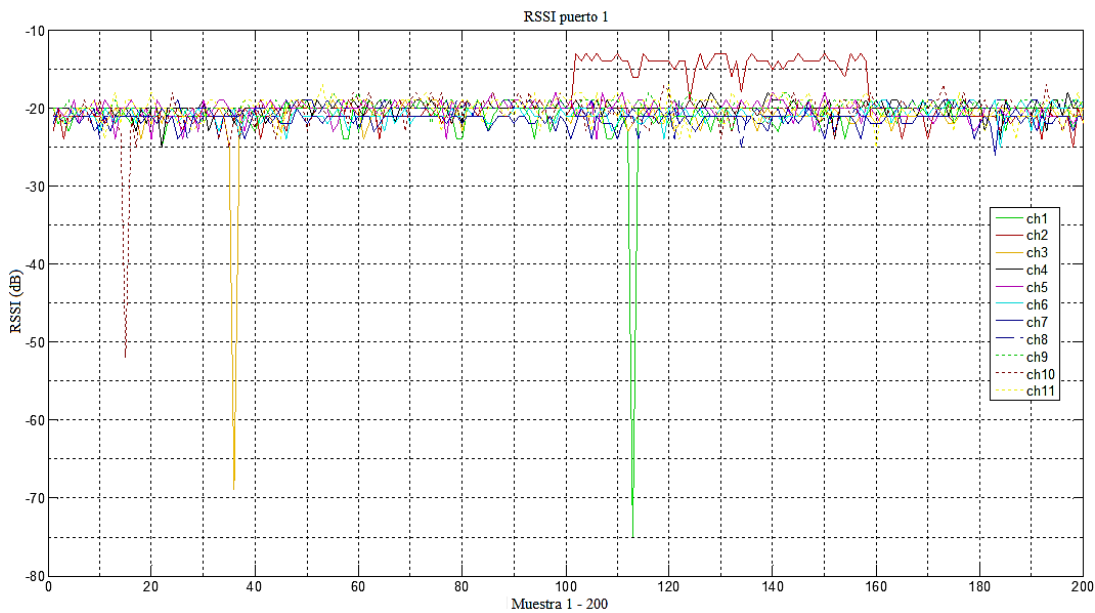


Ilustración 21. Medidas de RSSI puerto1.



Tanto en la ilustración anterior como en la siguiente, podemos hacer dos declaraciones. La primera y más importante es que tenemos la certeza del buen funcionamiento de la antena y el router, los valores son los esperados. La segunda sería la observación de algunos valores atípicos, aunque no se tiene muy clara la naturaleza de este comportamiento, se podría deducir que la intromisión de campo digital en estas pruebas provoca estas anomalías, ya que, en las pruebas analógicas, esto no ocurre.

Para descartar la mayor parte de errores por parte de la comunicación digital, se desactivó el control de potencia del router como se aclaró anteriormente, ya que éste podría nivelar la comunicación cuando fuese defectuosa.

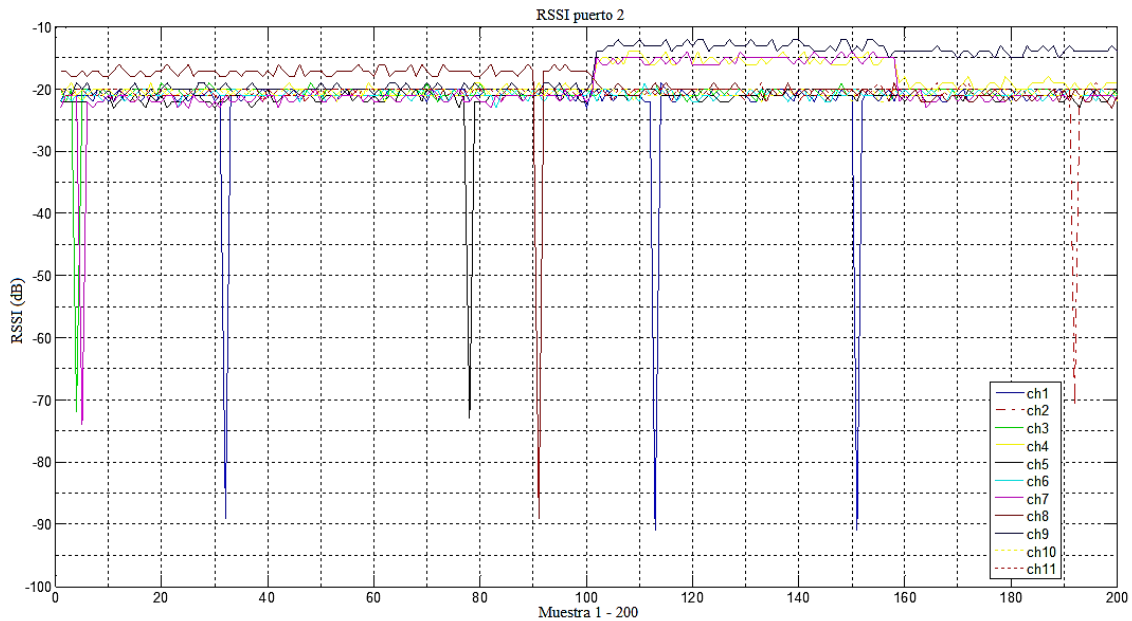


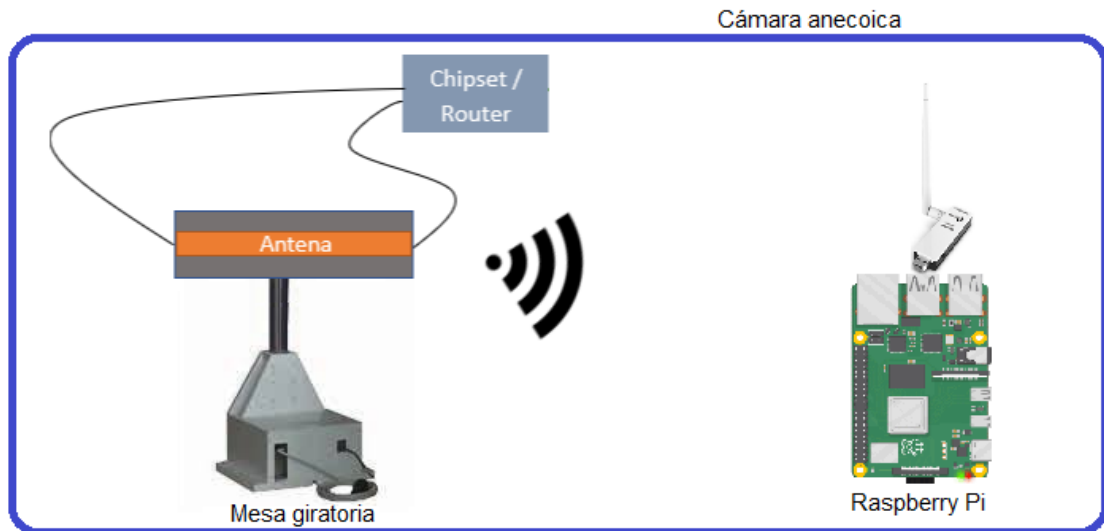
Ilustración 22. Medidas de RSSI puerto2.

También podemos observar que, en algunos momentos de la comunicación, la potencia recibida varía y se mantiene un tiempo determinado, como es el caso del canal 2 en la *Ilustración 20*, o los canales 4, 7 y 9 en la *Ilustración 21*. Esto se deberá tener en cuenta para el segundo escenario, ya que es el que realmente nos interesa. Se va a tratar los datos anómalos y los valores de comunicación cambiantes para poder estudiar mejor el comportamiento.

### 3.2. Pruebas experimentales: segundo escenario.

Nuestro objetivo ahora es el de obtener los diagramas digitales de nuestra antena, cuya comunicación viene regida por el router del primer escenario, con los diagramas digitales de la antena.

Con este fin, se van a tomar muestras del valor de RSSI de cada canal Wi-Fi y de cada dos ángulos de una barrida de 180 grados, desde  $-90^\circ$  a  $90^\circ$ . El router mandará los datos en bruto por UDP al portátil mientras los almacena en un archivo, después éste es enviado también bajo el mismo protocolo para poder así liberar la escasa memoria de la que dispone.



*Ilustración 23. Escenario de laboratorio 2.*

### 3.2.1. Descripción de componentes.

En este escenario se conserva gran parte de los dispositivos del apartado 3.1.1. con la salvedad de que ahora el ordenador portátil, aunque no cambia, hace también la función de girar la mesa posicionadora durante el experimento. Vamos a ver los nuevos componentes que entran en juego: ahora el router tendrá cada puerto conectado a uno de los dos que posee la antena Leaky Wave, la antena se colocará en la mesa posicionadora, la Raspberry hace su función como receptor inalámbrico y todo ello estará dentro de la cámara anecoica.

### Cámara anecoica

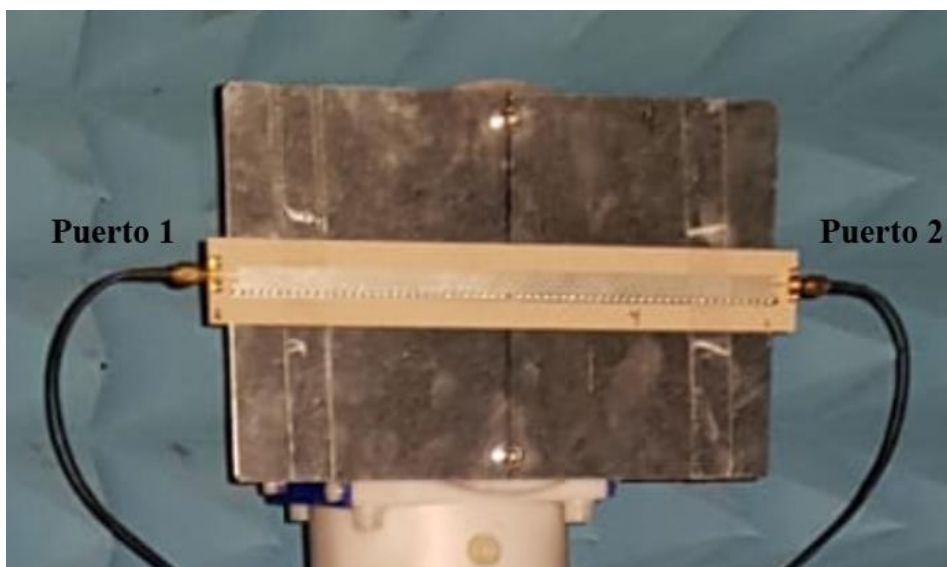
Es un espacio cerrado diseñado de forma que absorba todas las ondas acústicas y electromagnéticas en todo su interior y, por tanto, cualquier efecto que la propagación de ellas provoque. Ésta se encuentra aislada del exterior. La combinación de estas propiedades simula un espacio ideal para la propagación de las ondas.



*Ilustración 24. Cámara anecoica.*

La parte fundamental de la cámara anecoica son los revestimientos de suelo, paredes y techo, los cuales están hechos de un material que cumpla las propiedades de absorción que se necesitan, mayormente espumas o fibra de vidrio.

## **Antena Leaky Wave**



*Ilustración 25. Antena Leaky Wave.*

Esta antena se encuentra conectada con cada puerto en una de las dos conexiones de nuestro router. La peculiaridad de este tipo de antena es su efecto de fuga de onda, producido cuando parte de la onda electromagnética sale de la superficie. De forma que, cuando se alimente cada puerto, esta antena irradiará en distintos haces de la siguiente forma:

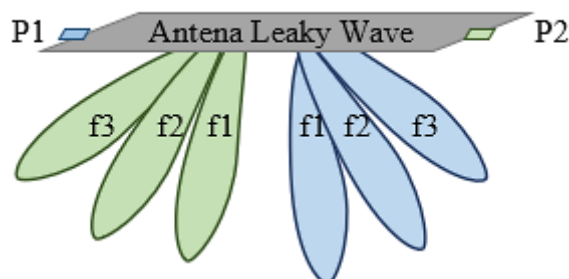


Ilustración 26. Radiación de antena Leaky Wave.

## Mesa giratoria Compact Table CT0800

La mesa giratoria está situada en uno de los extremos de la cámara anecoica, su función es la de orientar la antena durante las distintas pruebas.

Para trabajar con ella de forma manual es necesario un controlador, pero en nuestro caso utilizaremos una rutina de MATLAB para cambiar de ángulo cuando todas las muestras se hayan recogido. Para la comunicación entre la mesa y el controlador se utiliza fibra óptica y, entre la mesa y el ordenador, los conectores GPIB.



Ilustración 27. Mesa posicionadora Compact Table CT0800.

Características de la mesa posicionadora Compact Table CT0800.	
Diámetro	800 mm
Carga admisible	máx. 75kg
Altura	188 mm
Placa portadora de material	Kömacel, PVC-borde-anillo, resistente a la intemperie o madera
Min. Tiempo de rotación	0,5 rpm.
Max. Tiempo de rotación	2,0 rpm.
Precisión de posicionamiento	+/- 1 °
Ángulo de rotación	± 200 ° o -20 ° a + 380 °
Controlar	cable de fibra óptica (tipo POF)
Motor	Motor EC electrónico 150W
Unidad de manejo	blindado y radiointerferencia suprimida; 20dB bajo CISPR 22 Clase B
Rango de temperatura	0 ° C ... + 40 ° C
Rango de trabajo	5 ° C ... + 35 ° C
Consumo actual	max. 1,6A
Voltaje	115 / 230V (50-60Hz)

Tabla 7. Características de la mesa posicionadora Compact Table CT0800.

## Script para el control del router

Para comprender en totalidad el experimento vamos a ver cómo hemos programado el script sobre Bash.

En primer lugar, se declaró un bucle para contener toda la funcionalidad que depende del ángulo en el que nos encontramos, para el script se trata del número de ángulos a contar como variable de incremento. Para el barrido de canales se utiliza el formato de sierra, es decir, vamos a recorrerlos del 11 al 1 y, acto seguido, del 1 al 11. Esta distribución se escogió para no forzar al router a hacer cambios bruscos en la frecuencia de canal, ya que producía variaciones de potencia de señal muy drásticas.

Luego se formularon tres comprobaciones, aunque se prescindió de la última de ellas tras ver que las dos primeras eran suficiente. Primero se comprueba el canal, de forma que la frecuencia a la que se le ha obligado a cambiar sea la misma que en la que se encuentra en ese momento. En la segunda comprobación se observa si el tiempo de conexión, es decir, el tiempo que lleva conectado el dispositivo receptor a ese canal es menor que el tiempo de la anterior trama recogido, si esto se cumple sabemos que el cambio de canal se ha producido de manera satisfactoria. La tercera comprobación consistía en lanzar un ping de emisor a receptor y comprobar si existía respuesta, aspecto que se eliminó al experimentar con el funcionamiento del comando utilizado para la absorción de datos.

Una vez todas las comprobaciones se han realizado, se prepara la primera recogida de datos para la primera trama, se recoge el canal actual, el tiempo de conexión y los datos de la comunicación. Para los datos se utiliza el comando `iw dev wlan1 station get d0:37:45:55:6e:a9`, donde se especifica la MAC del receptor con el que nos comunicamos. De este comando podemos recibir gran cantidad de datos, pero nosotros nos quedaremos con las tasas de envío y recepción, el ancho de banda esperado y el valor de RSSI. Cuando los datos que se han comentado se han enviado por UDP, se guardan junto al resto en un archivo de texto para su posterior consulta en caso de ser necesario. Como se dijo anteriormente, al cambiar de ángulo, el archivo de texto con estos datos se envía por UDP al mismo portátil.

### 3.2.2. Resultados.

Una vez la prueba ha terminado, se deben procesar los datos para eliminar los fallos comentados en el apartado 3.1.2, de forma que descartemos los valores atípicos sustituyéndolos por la media del resto y suavizando la curva comparando los valores discretos de cada ángulo con los de sus vecinos más próximos. Además, se utilizó la función *smooth* de MATLAB mediante el método de *Savitzky Golay*.

Así, se pudo obtener una gráfica que podemos comparar fácilmente con los diagramas analógicos.

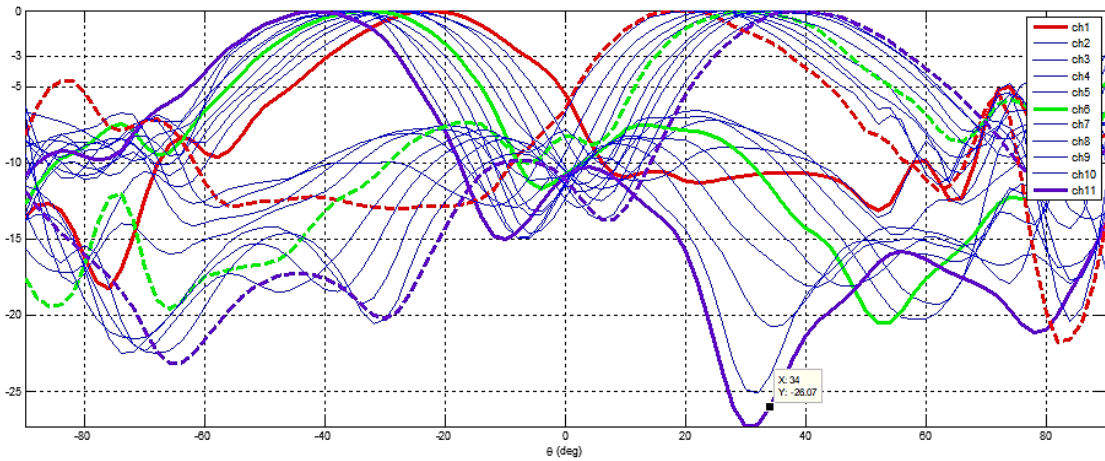


Ilustración 28. Diagrama de radiación digital de la antena.

En la *Ilustración 27*, podemos ver el diagrama de radiación digital de la antena Leaky Wave de ambos puertos.

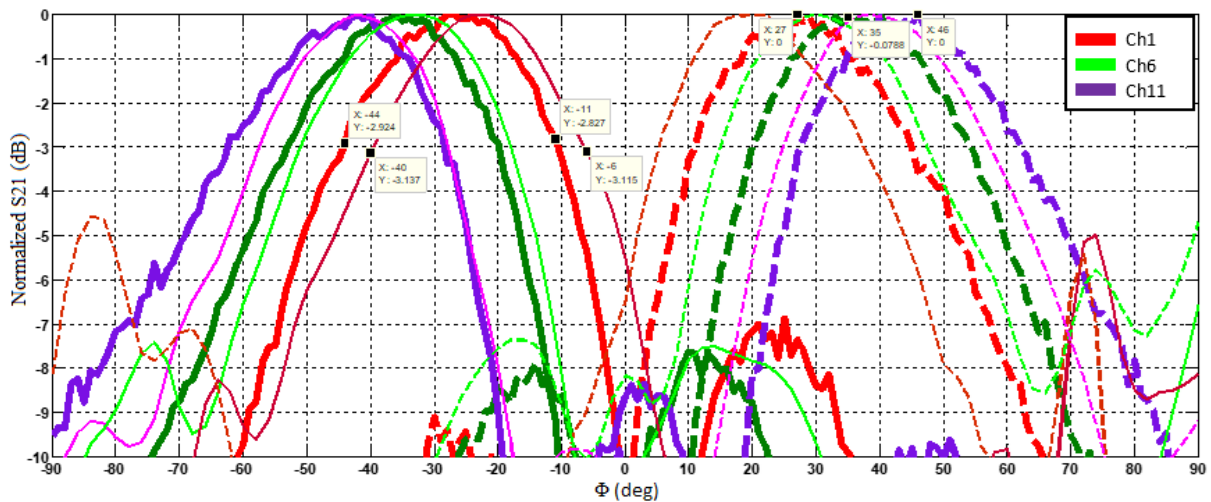


Ilustración 29. Comparativa diagramas de radiación analógico y digital.

Ahora observamos ambos diagramas normalizados, donde las líneas más gruesas corresponden con los datos analógicos y, las más finas, con los datos digitales. Vemos que la superposición es correcta a pesar de no ser exacta. Para ambos casos existe una zona de punto ciego o sombra en el centro, donde lo ideal sería que su cruce en cero se produjese a valores mayores de RSSI en la gráfica.

Esto da por satisfecha la toma de medidas, ahora sabemos que podemos utilizar la señal inalámbrica Wi-fi para intentar localizar un dispositivo o favorecer una comunicación mediante el uso de parámetros de calidad de servicio, los cuáles nos permitirán conocer la perpendicularidad de un dispositivo conectado a una frecuencia concreta respecto del haz de dicho canal.

# Capítulo 4: Automatización del proceso en diferentes escenarios.

## 4.1. Estudio de parámetros de QoS.

Hasta este punto hemos estudiado sólo uno de los parámetros que estábamos recogiendo de cada muestra, ahora debemos centrarnos en los parámetros que condicionan la QoS de la comunicación. Vamos a fijar una serie de reglas para poder realizar un salto en frecuencia de forma automática a través del estudio de estos parámetros.

Antes de nada, debemos comprender lo que significa QoS. Se trata de un mecanismo usado para poder reglar la priorización del tráfico y asegurar un ancho de banda mínimo para una calidad de comunicación determinada. Nosotros escogeremos como parámetros esenciales el throughput (rendimiento, en español) y el número de paquetes perdidos en transporte [17]. A su vez se define throughput como la cantidad de información transmitida de forma correcta en una comunicación o, también como la velocidad de transferencia de la red, siempre menor que el ancho de banda del canal [18].

Durante el experimento en la cámara anecoica fuimos almacenando en archivos los datos en bruto de toda la comunicación y es con ellos con quienes fijaremos nuestras reglas.

```
root@(none):~# iw dev wlan1 station get d0:37:45:55:6e:a9
Station d0:37:45:55:6e:a9 (on wlan1)
  inactive time: 1140 ms
  rx bytes: 56420
  rx packets: 598
  tx bytes: 49185
  tx packets: 386
  tx retries: 216
  tx failed: 11
  signal: -34 [-55, -40, -35] dBm
  signal avg: -48 [-71, -54, -50] dBm
  tx bitrate: 65.0 MBit/s MCS 7
  rx bitrate: 1.0 MBit/s
  expected throughput: 32.42Mbps
  authorized: yes
  authenticated: yes
  preamble: short
  WMM/WME: yes
  MFP: no
  TDLS peer: no
  connected time: 369 seconds
```

*Ilustración 30. Ejecución de comando iw station.*

En la *Ilustración 29* podemos ver todos los parámetros que podemos obtener de la comunicación. Según la orientación de la antena del emisor con respecto al receptor, los valores deseados varían, haciendo que cuando los haces del canal al que estamos conectados no miran directamente al receptor, sufrimos unas pérdidas de paquetes superiores al resto (tx failed en la salida del comando) y un valor de throughput (expected throughput en la salida de comando) también por debajo de la media.

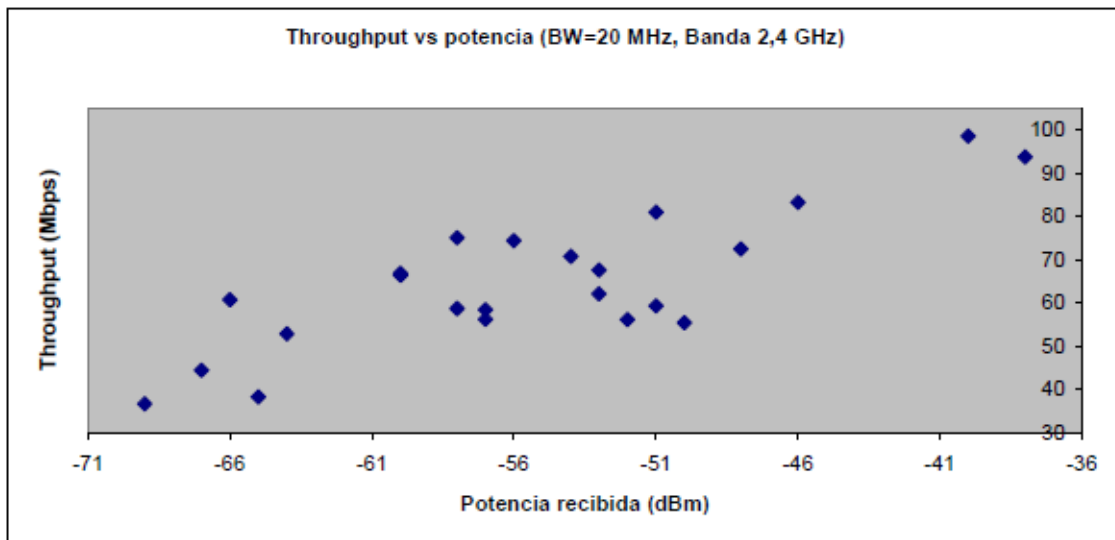
Se va a utilizar la ecuación de la *Ilustración 31* para fijar un umbral cuyo objetivo es maximizar la potencia tanto en el enlace de subida como el de bajada, pudiendo así tener la oportunidad de transmitir con el mejor MCS (Modulation and Coding Scheme) y por lo tanto maximizar el throughput [19]. En nuestro caso debemos de optar por un valor de *MargenRSSI* para que las transiciones entre canales no sean demasiado consecutivas.

$$RSSI_{AP\text{ objetivo}} [dB] \geq RSSI_{AP\text{ actual}} [dB] + Margen_{RSSI} [dB]$$

*Ilustración 31. Ecuación de reasociación para niveles de throughput*

El valor de MCS define cuántos bits útiles se transmiten hacia el receptor. Las tasas más elevadas tienen más datos de usuario y menos redundancia, encontrándonos con la desventaja de una recuperación ante fallos menos eficiente ya que éstos se dan menos[20].

Además, si comparamos las potencias recibidas con los niveles de throughput obtenidos a lo largo de una prueba donde varía el valor de RSSI, podemos observar que, observando la *Ilustración 29*, junto a la siguiente imagen, podemos fijar el nivel de throughput mínimo para un cambio de canal en el 50%.



*Ilustración 32. Throughput vs potencia recibida [21].*

La pérdida de paquetes se considera una medida más crítica que el throughput y, por tanto, se le dará prioridad a la hora de realizar la automatización del proceso. Por ello, se va a medir el valor de Data Rate correspondiente al MCS deseado según el estándar 802.11ac, estos valores se reflejan en la *Ilustración 33*. El valor máximo de MCS que puede contemplar nuestra antena según los valores recogidos en el primer experimento puede ser de 7, al cruce de canales observados en la *Ilustración 29* estaríamos en un nivel 5 de MCS y es el valor que utilizaremos como umbral para controlar el cambio de canal y evitar así pérdidas de comunicación.



VHT MCS	Modulation	Coding	20MHz				40MHz				80MHz				160MHz			
			Data Rate		Min. SNR	RSSI	Data Rate		Min. SNR	RSSI	Data Rate		Min. SNR	RSSI	Data Rate		Min. SNR	RSSI
			800ns	400ns			800ns	400ns			800ns	400ns			800ns	400ns		
0	BPSK	1/2	6.5	7.2	2	-82	13.5	15	5	-79	29.3	32.5	8	-76	58.5	65	11	-73
1	QPSK	1/2	13	14.4	5	-79	27	30	8	-76	58.5	65	11	-73	117	130	14	-70
2	QPSK	3/4	19.5	21.7	9	-77	40.5	45	12	-74	87.8	97.5	15	-71	175.5	195	18	-68
3	16-QAM	1/2	26	28.9	11	-74	54	60	14	-71	117	130	17	-68	234	260	20	-65
4	16-QAM	3/4	39	43.3	15	-70	81	90	18	-67	175.5	195	21	-64	351	390	24	-61
5	64-QAM	2/3	52	57.8	18	-66	108	120	21	-63	234	260	24	-60	468	520	27	-57
6	64-QAM	3/4	58.5	65	20	-65	121.5	135	23	-62	263.3	292.5	26	-59	526.5	585	29	-56
7	64-QAM	5/6	65	72.2	25	-64	135	150	28	-61	292.5	325	31	-58	585	650	34	-55
8	256-QAM	3/4	78	86.7	29	-59	162	180	32	-56	351	390	35	-53	702	780	38	-50
9	256-QAM	5/6			31	-57	180	200	34	-54	390	433.3	37	-51	780	866.7	40	-48

Ilustración 33. Índice de MCS para el estándar 802.11ac [22].

## 4.2. Desarrollo de automatización.

La automatización se ha programado sobre el lenguaje anteriormente mencionado Bash. Como se mencionó en el apartado anterior, se han priorizado las pérdidas en comunicación sobre el rendimiento del canal. Para ello, se comprueba inicialmente que valor MCS asociado a las pérdidas se mayor de 4, si esto no se cumple, no se comprobará el throughput y se procederá a cambiar de canal. Podemos ver en la *Ilustración 34* que el diagrama de estados respeta esta premisa.

Si ambas comprobaciones han sido satisfactorias, estaríamos en situación de canal adecuado, es decir, la comunicación puede continuar sin problemas. Cuando estamos en esta situación, el programa vuelve a la rutina de comprobaciones tanto de pérdidas como de rendimiento, actualizando constantemente los valores de QoS y aumentando la confiabilidad de la conexión.

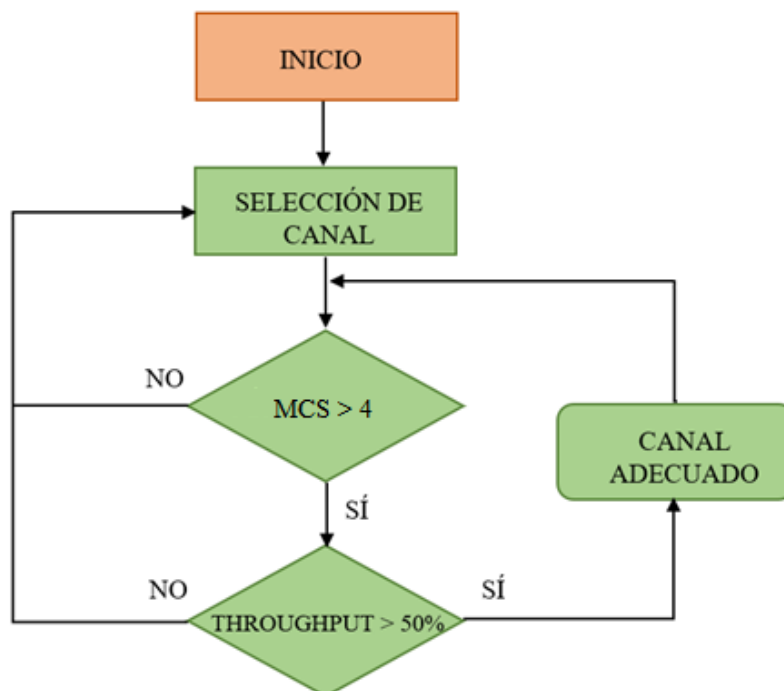


Ilustración 34. Flujo de estados del programa.

# Capítulo 5: Conclusión y líneas futuras.

## 5.1. Conclusión.

En este proyecto se ha buscado cumplir dos objetivos principales, el estudio de las técnicas de Frequency Hopping con el uso de una antena Leaky Wave y la creación de una aplicación que partiera de dicho estudio.

Para la realización de todo ello, primero se estudió los aspectos esenciales de la técnica de salto de canal, tales como la trama que contiene el aviso de cambio de canal, trama beacon, o quién entre emisor y receptor daba la orden de salto, concluyendo que era el emisor quien lo hacía. Entonces se implementó, mediante una placa embebida y un chipset inalámbrico, un router con el que gestionar y controlar el punto de acceso, la antena.

El router se controla mediante un script programado en lenguaje Bash con el que forzamos al cambio de canal y recogemos las muestras necesarias para el estudio, además se enviaban mientras tanto todos los datos en bruto almacenados en un archivo para su posterior uso, todo ello vía UDP. Cuando se recogieron los datos de las pruebas y se procesaron pudimos observar que los diagramas digitales y analógicos se correspondían, señal de que la antena había actuado correctamente en la comunicación W-Fi.

Una vez observamos que la antena es apta para la comunicación, pasamos a crear un script que dotase al router de la capacidad de cambiar de canal dependiendo de unos valores predefinidos, dichos valores corresponden a parámetros de QoS. Mediante la observación de los parámetros de pérdida de paquetes y el rendimiento o throughput, el router puede cambiar de canal para mejorar la comunicación y no perder datos.

## 5.2. Líneas futuras.

Uno de los posibles escenarios a estudiar sería el de la localización mediante este tipo de montaje desarrollado en el proyecto, refiriéndonos tanto a localización indoor como en el exterior. Es interesante la idea de uso en las flying networks, es decir, posicionar los dispositivos a bordo de un dron y poder utilizar las características de potencia y de QoS en conjunto para poder localizar dispositivos en función de éstas. Sabemos mediante el procesado de las potencias que la dirección de los haces podría ser determinante a la hora de saber dónde se encuentra el receptor.

Continuando con la línea de las flying networks y en relación con el capítulo 4 de este informe, donde se realiza una aplicación para el mantenimiento de una correcta comunicación cuando se presupone un objetivo en movimiento, también se podría plantear un dron o conjunto de drones como punto de acceso para eventos multitudinarios donde actualmente es necesario desplazar infraestructura móvil de telecomunicaciones para paliar las deficiencias de una red sobrecargada.

Otra vertiente que apareció durante la realización del proyecto fue la de estrechar el ancho de transmisión, obligados por defecto a su uso en 20MHz debido al modelo de placa que estábamos usando. Con otros dispositivos que nos permitan estrechar este valor podríamos asemejar mejor los diagramas de radiación digital a los analógicos.



## Bibliografía

- [1] J. Andreu, "Redes inalámbricas (Servicios en red)," vol. 53, no. 9, 2019.
- [2] A. I. Al-Alawi, "WiFi Technology: Future Market Challenges and Opportunities," *J. Comput. Sci.*, vol. 2, no. 1, pp. 13–18, 2006.
- [3] R. Xu, G. Shi, J. Luo, Z. Zhao, and Y. Shu, "MuZi: Multi-channel ZigBee networks for avoiding WiFi interference," *Proc. - 2011 IEEE Int. Conf. Internet Things Cyber, Phys. Soc. Comput. iThings/CPSCOM 2011*, pp. 323–329, 2011.
- [4] "Tipos de redes y estándares Wi-Fi, sus diferencias," pp. 2–6, 2019.
- [5] O. H. Us, L. Meredith, and E. J. Freeman, "Multiple access frequency hopping network with interference anticipation," vol. 2, no. 12, 2013.
- [6] H. Córdova and P. Chávez, "Estudio , Modelamiento y Simulación de Sistemas de Espectro Ensanchado Secuencia Directa y Salto De Frecuencia," vol. 1, pp. 1–8, 2005.
- [7] N. Hossein, "Frequency Hopping Spread Spectrum: An Effective Way to Improve Wireless Communication Performance," *Adv. Trends Wirel. Commun.*, no. February 2011, 2011.
- [8] E. O. Matthew S. Gast, *Beamforming in 802.11ac*. 2020.
- [9] "802.11 ac wave 2, Beamforming y MU-MIMO," pp. 3–7, 2020. <https://naseros.com/2018/11/05/802-11-ac-wave-2-beamforming-y-mu-mimo/>
- [10] SIMON, "La guía definitiva para entender el cable coaxial," p. 31, 2019.
- [11] "Tipos de cableado Guiados Guiados : Fibra optica," pp. 2–4, 2020.
- [12] Bash - GNU Project - Free Software Foundation: <https://www.gnu.org/software/bash/>
- [13] Descripción de MATLAB, MathWorks España: [https://es.mathworks.com/help/matlab/learn\\_matlab/product-description.html](https://es.mathworks.com/help/matlab/learn_matlab/product-description.html)
- [14] WPJ531 Reliable QCA9531 650Mhz 2X2 MIMO 802.11n Wireless Embedded Board: <http://shop.compex.com.sg/embedded-boards/wpj531lv.html>
- [15] OpenWrt Project\_ Compex WPJ531: <https://openwrt.org/toh/compex/wpj531>
- [16] ¿Qué es una cámara anecoica\_ - Aislacustic Ingeniería Acústica: <https://aislacustic.com/que-es-una-camara-anecoica/>
- [17] M.R. Cruz Felipe, R. Martínez Gómez, Y. Crespo García. "Análisis de la QoS en redes inalámbricas." *Rev. cuba cienc. informat.* vol.7 no.1 La Habana ene.-mar. 2013
- [18] F. F. Pérez, "Trabajo Fin de Grado ( Deployment of a platform for analysis of network techniques coding using Raspberry Pi ) Graduado en Ingeniería de Tecnologías de Telecomunicación," 2017.
- [19] A. M. Pascual, "BALANCE DE TRÁFICO Y PROVISIÓN DE QOS EN REDES IEEE 802.11," *Zaguan.Unizar.Es*, pp. 0–43, 2014.
- [20] V. J. María, "Estudio de las técnicas de Beamforming en el estándar 802.11ac," p. 134, 2015.
- [21] L. Cano Pérez, "CONFIGURACIÓN Y EVALUACIÓN DE REDES 802.11n," vol. 1, p. 80, 2013.
- [22] MCS index charts \_ Wireless LAN Professionals: <https://www.wlanpros.com/mcs-index-charts/>

## ANEXO

### A. Medición 11 canales Wi-Fi en ambos puertos de la antena, en 180 grados de giro. (fhop11.sh)

```
#!/bin/bash

rm lecturapotencias.txt
t_before='10000'
start_time=$(date +%s)

for angulo in `seq 1 10`
do
sleep 2

for i in 2462 2457 2452 2447 2442 2437 2432 2427 2422 2417 2412 2412 2417 2422 2427
2432 2437 2442 2447 2452 2457 2462
do

    echo '-----'
    echo "Changing channel to ch: $i"
    hostapd_cli -i wlan1 chan_switch 1 $i # JK: para cambiar de canal: chan_switch. usage:
<cs_count> <freq> [sec_channel_offset=] [center_freq1=]
# [center_freq2=] [bandwidth=] [blocktx]
[ht|vht]
    num="$i" #JK ¿Es necesaria esta variable? Con $i debería servir, supongo.

#COMPROBACION DEL CANAL
    while :
    do
        chan=$(iw dev | grep 'channel' |awk '{print $11;}')
        if [ $num = $chan ]; then
            echo "Changed Channel to ch $chan"
            break
        fi
    done

#COMPROBACION DE TIEMPO CONECTADO

    while :
    do
        t_connected=$(iw dev wlan1 station get d0:37:45:55:6e:a9 | grep 'connected time'|awk
'{'print $3;}')
        if [ $t_connected -lt $t_before ]; then
            echo "Time ok"
            break
        fi
    done

    for j in `seq 1 10` #JK: para hacer las 10 medidas.
do
```

```

#COMPROBACION DEL DISPOSITIVO CONECTADO
#   while :
#   do
#       check_ping=`ping -c 1 192.168.2.101`
#       check=$? #devuelve 0 si el ping ha llegado y 1 si no llega
#       zero="0"
#       if [ $check = $zero ]; then
#       echo "Device connected"
#       break
#       fi
#   done

#MEDIDA DE POTENCIA Y ENVIO DE DATOS
chan=$(iw dev | grep 'channel' |awk '{print $11;}')
echo "Canal Actual: $chan"

if [ $j = 1 ]; then
    finish_time=$(date +%s)
    echo "Time duration: $((finish_time-start_time)) secs."
fi

if [ $j = 10 ]; then
    start_time=$(date +%s)
    fi

    data=$(iw dev wlan1 station get d0:37:45:55:6e:a9)

    pot=$(echo $data | awk '{print $27, $28, $29, $30, $31;}')
    echo $pot
    txbitrate=$(echo $data | awk '{print $40,$41,$42,$43;}')
    echo $txbitrate
    rxbitrate=$(echo $data | awk '{print $44,$45,$46,$47;}')
    echo $rxbitrate
    expected=$(echo $data | awk '{print $48,$49,$50;}')
    bad="authorized: yes authenticated:"

    if [ "$expected" = "$bad" ]; then
        expected="expected throughput: NaN"
        echo $expected
    else
        echo $expected
    fi

    echo

#medidas auxiliares

    echo "Canal: $chan Muestra: $j $data" >> lecturapotencias.txt
#   iw dev wlan1 station dump get d0:37:45:55:6e:a9 >> lecturapotencias.txt
    echo >> lecturapotencias.txt

    paq_udp="Channel: ${chan} ${pot}"
    echo $paq_udp | stdbuf -o 0 socat - udp-sendto:192.168.55.101:1883

```

```

                #sleep 1

done

    sleep 3
    lim1="2412"
    lim2="2462"
    if [ $i = $lim1 ] || [ $i = $lim2 ]; then
        t_before='100000'
    else
        t_before=$(iw dev wlan1 station get d0:37:45:55:6e:a9 | grep 'connected time'|awk
'{{print $3;}}')
    fi

#    echo $t_before
#    echo $t_connected
done # for channel
echo
done # for angulo

```

## B. Toma de muestras en la cámara anecoica (medidas\_rssi\_wifi.m)

```

%Función para tomar medidas de la RSSI

close all;
clear all;
clc;

n=100; %Número de muestras por canal
ang=91; %Número de angulos para tomar medidas

port=1883; %Puerto para las potencias
timeout=10000;
ip_add='192.168.55.82'; %dirección ip de donde recibe los UDP (router)
data=udp(ip_add,'LocalPort',port,'Timeout',timeout); %definicion del
paquete udp

posp1=1;%valores para seleccionar la posicion en el dibujo
posp2=2;
VECTOR_P1= zeros([n,22,ang]); %vectores que guardan una matriz de
medidas por cada angulo
VECTOR_P2= zeros([n,22,ang]);
VECTOR_P1_MEDIOS_11=zeros([ang,11]);
VECTOR_P2_MEDIOS_11=zeros([ang,11]);
VECTOR_P1_MEDIOS_11(:,:)= -100;
VECTOR_P2_MEDIOS_11(:,:)= -100;

VECTOR_P1_raw= zeros([n,22,ang]); %medidas sin procesar
VECTOR_P2_raw= zeros([n,22,ang]);
figure('Name','CARACTERIZACIÓN ANTENA','NumberTitle','off');
    set(gcf, 'Position', [400, 200, 1000, 600])

legend('ch1','ch2','ch3','ch4','ch5','ch6','ch7','ch8','ch9','ch10','c
h11')
    title('Potencia en dBm para 11 canales en función del ángulo')
    xlim([1 ang-1]);

```

```

XTick = 1:2.5:ang-1;
set(gca,'xtick',XTick)
set(gca,'xticklabel',({'-90','-85','-80','-75','-70','-65','-60','-55','-50','-45','-40','-35','-30','-25','-20','-15','-10','-5','0','+5','+15','+20','+25','+30','+35','+40','+45','+50','+55','+60','+65','+70','+75','+80','+85','+90'}))
xlabel('Ángulo Theta')
ylabel('Potencia (dBm)')
grid on
%%variables de simulacion
%%Configuración de mesa
step=2;
angulo_inicio=90;
angulo_giro= 270;
correctang=angulo_inicio-1; %para rellenar bien las matrices
a=1;b=1;c=1;d=1;e=1;f=1;g=1;h=1;k=1;l=1;m=1;
retardo=1;

mesa = visa ('agilent', 'GPIB0::7::INSTR' );

% % Abrimos comunicacion
fopen(mesa);

fprintf(mesa,'LD 1 DV'); % seleccionamos la mesa giratoria
fprintf(mesa,['LD' num2str(angulo_inicio) ' DG NP GO']) ;
pause(retardo);

%Bucle de giro

theta=1;
for theta2=angulo_inicio:step:angulo_giro

    disp(['Cambio de grados ',num2str(theta2)]);
    fprintf(mesa,'LD 1 DV'); % Seleccionamos la mesa giratoria
    fprintf(mesa,['LD ' num2str(theta2) ' DG NP GO']);

    pause(retardo);

for pasada=1:2
    a=1;b=1;c=1;d=1;e=1;f=1;g=1;h=1;k=1;l=1;m=1;

for i=1:11*n

    if (a>n && pasada==1)
        break;
    end

    if(m>n && pasada ==2)
        break;
    end

    VECTOR_P1(:, :, theta)
    canales=[' ch1' ' ch2' ' ch3' ' ch4' ' ch5' ' ch6'
' ch7' ' ch8' ' ch9' ' ch10' ' ch11' ' ch11' ' ch10' '
ch9' ' ch8' ' ch7' ' ch6' ' ch5' ' ch4' ' ch3' ' ch2' '
ch1']

    disp(theta2)

```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%LEEMOS TRAMAS UDP%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fopen(data);
Potencias=fread(data);
char(Potencias')
fclose(data);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

tic
long=length(Potencias');
if (long<42 || char(Potencias(23))==0)
%comprobacion en caso de desconexión o potencia 0

    if((length(Potencias)==1) && (i>50) && (char(Potencias)=='C'))
%comprobación de seguridad en caso de que la lectura de matlab de la
trama UDP falle.
        continue;
    end

    if str2double(char(Potencias(10:13)'))==2412
%%El programa pone potencia -100 si ha habido desconexión o la
potencia recibida es 0dBm
        if pasada==1
            VECTOR_P1(a,1,theta)=-100;
            VECTOR_P2(a,1,theta)=-100;
            a=a+1;
        elseif pasada==2
            VECTOR_P1(a,22,theta)=-100;
            VECTOR_P2(a,22,theta)=-100;
            a=a+1;
        end
    elseif str2double(char(Potencias(10:13)'))==2417
        if pasada==1
            VECTOR_P1(b,2,theta)=-100;
            VECTOR_P2(b,2,theta)=-100;
            b=b+1;
        elseif pasada==2
            VECTOR_P1(b,21,theta)=-100;
            VECTOR_P2(b,21,theta)=-100;
            b=b+1;
        end
    elseif str2double(char(Potencias(10:13)'))==2422
        if pasada==1
            VECTOR_P1(c,3,theta)=-100;
            VECTOR_P2(c,3,theta)=-100;
            c=c+1;
        elseif pasada==2
            VECTOR_P1(c,20,theta)=-100;
            VECTOR_P2(c,20,theta)=-100;
            c=c+1
        end
    elseif str2double(char(Potencias(10:13)'))==2427
        if pasada==1
            VECTOR_P1(d,4,theta)=-100;
            VECTOR_P2(d,4,theta)=-100;
            d=d+1;
        elseif pasada==2
            VECTOR_P1(d,19,theta)=-100;
            VECTOR_P2(d,19,theta)=-100;
            d=d+1
        end
    elseif str2double(char(Potencias(10:13)'))==2432

```

```

if pasada==1
    VECTOR_P1(e,5,theta)==-100;
    VECTOR_P2(e,5,theta)==-100;
    e=e+1;
elseif pasada==2
    VECTOR_P1(e,18,theta)==-100;
    VECTOR_P2(e,18,theta)==-100;
    e=e+1;
end
elseif str2double(char(Potencias(10:13)))==2437
    if pasada==1
        VECTOR_P1(f,6,theta)==-100;
        VECTOR_P2(f,6,theta)==-100;
        f=f+1;
    elseif pasada==2
        VECTOR_P1(f,17,theta)==-100;
        VECTOR_P2(f,17,theta)==-100;
        f=f+1;
    end
elseif str2double(char(Potencias(10:13)))==2442
    if pasada==1
        VECTOR_P1(g,7,theta)==-100;
        VECTOR_P2(g,7,theta)==-100;
        g=g+1;
    elseif pasada==2
        VECTOR_P1(g,16,theta)==-100;
        VECTOR_P2(g,16,theta)==-100;
        g=g+1;
    end
elseif str2double(char(Potencias(10:13)))==2447
    if pasada==1
        VECTOR_P1(h,8,theta)==-100;
        VECTOR_P2(h,8,theta)==-100;
        h=h+1;
    elseif pasada==2
        VECTOR_P1(h,15,theta)==-100;
        VECTOR_P2(h,15,theta)==-100;
        h=h+1;
    end
elseif str2double(char(Potencias(10:13)))==2452
    if pasada==1
        VECTOR_P1(k,9,theta)==-100;
        VECTOR_P2(k,9,theta)==-100;
        k=k+1;
    elseif pasada==2
        VECTOR_P1(k,14,theta)==-100;
        VECTOR_P2(k,14,theta)==-100;
        k=k+1;
    end
elseif str2double(char(Potencias(10:13)))==2457
    if pasada==1
        VECTOR_P1(l,10,theta)==-100;
        VECTOR_P2(l,10,theta)==-100;
        l=l+1;
    elseif pasada==2
        VECTOR_P1(l,13,theta)==-100;
        VECTOR_P2(l,13,theta)==-100;
        l=l+1;
    end
elseif str2double(char(Potencias(10:13)))==2462
    if pasada==1

```

```

        VECTOR_P1(m,11,theta)=-100;
        VECTOR_P2(m,11,theta)=-100;
        m=m+1;
        elseif pasada==2
            VECTOR_P1(m,12,theta)=-100;
            VECTOR_P2(m,12,theta)=-100;
            m=m+1;
        end

    end

elseif str2double(char(Potencias(10:13)))==2412
    if pasada==1
        VECTOR_P1(a,1,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(a,1,theta)=str2double(char(Potencias(38:40)));
        a=a+1;
    elseif pasada==2
        VECTOR_P1(a,22,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(a,22,theta)=str2double(char(Potencias(38:40)));
        a=a+1;
    end
elseif str2double(char(Potencias(10:13)))==2417
    if pasada==1
        VECTOR_P1(b,2,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(b,2,theta)=str2double(char(Potencias(38:40)));
        b=b+1;
    elseif pasada==2
        VECTOR_P1(b,21,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(b,21,theta)=str2double(char(Potencias(38:40)));
        b=b+1;
    end
elseif str2double(char(Potencias(10:13)))==2422
    if pasada==1
        VECTOR_P1(c,3,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(c,3,theta)=str2double(char(Potencias(38:40)));
        c=c+1;
    elseif pasada==2
        VECTOR_P1(c,20,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(c,20,theta)=str2double(char(Potencias(38:40)));
        c=c+1;
    end
elseif str2double(char(Potencias(10:13)))==2427
    if pasada==1
        VECTOR_P1(d,4,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(d,4,theta)=str2double(char(Potencias(38:40)));
        d=d+1;
    elseif pasada==2
        VECTOR_P1(d,19,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(d,19,theta)=str2double(char(Potencias(38:40)));
        d=d+1;
    end
elseif str2double(char(Potencias(10:13)))==2432
    if pasada==1
        VECTOR_P1(e,5,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(e,5,theta)=str2double(char(Potencias(38:40)));
        e=e+1;
    elseif pasada==2
        VECTOR_P1(e,18,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(e,18,theta)=str2double(char(Potencias(38:40)));
        e=e+1;
    end
end

```

```

elseif str2double(char(Potencias(10:13)))==2437
    if pasada==1
        VECTOR_P1(f,6,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(f,6,theta)=str2double(char(Potencias(38:40)));
        f=f+1;
    elseif pasada==2
        VECTOR_P1(f,17,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(f,17,theta)=str2double(char(Potencias(38:40)));
        f=f+1;
    end
elseif str2double(char(Potencias(10:13)))==2442
    if pasada==1
        VECTOR_P1(g,7,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(g,7,theta)=str2double(char(Potencias(38:40)));
        g=g+1;
    elseif pasada==2
        VECTOR_P1(g,16,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(g,16,theta)=str2double(char(Potencias(38:40)));
        g=g+1;
    end
elseif str2double(char(Potencias(10:13)))==2447
    if pasada==1
        VECTOR_P1(h,8,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(h,8,theta)=str2double(char(Potencias(38:40)));
        h=h+1;
    elseif pasada==2
        VECTOR_P1(h,15,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(h,15,theta)=str2double(char(Potencias(38:40)));
        h=h+1;
    end
elseif str2double(char(Potencias(10:13)))==2452
    if pasada==1
        VECTOR_P1(k,9,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(k,9,theta)=str2double(char(Potencias(38:40)));
        k=k+1;
    elseif pasada==2
        VECTOR_P1(k,14,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(k,14,theta)=str2double(char(Potencias(38:40)));
        k=k+1;
    end
elseif str2double(char(Potencias(10:13)))==2457
    if pasada==1
        VECTOR_P1(l,10,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(l,10,theta)=str2double(char(Potencias(38:40)));
        l=l+1;
    elseif pasada==2
        VECTOR_P1(l,13,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(l,13,theta)=str2double(char(Potencias(38:40)));
        l=l+1;
    end
elseif str2double(char(Potencias(10:13)))==2462
    if pasada==1
        VECTOR_P1(m,11,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(m,11,theta)=str2double(char(Potencias(38:40)));
        m=m+1;
    elseif pasada==2
        VECTOR_P1(m,12,theta)=str2double(char(Potencias(33:35)));
        VECTOR_P2(m,12,theta)=str2double(char(Potencias(38:40)));
        m=m+1;
    end
% elseif str2double(char(Potencias(10:13)))==2467

```

```

%         ch12_p1(p)=str2double(char(Potencias(70:72)'));
%         ch12_p2(p)=str2double(char(Potencias(75:77)'));
%         p=p+1
%     elseif str2double(char(Potencias(10:13)'))==2472
%         ch13_p1(o)=str2double(char(Potencias(70:72)'));
%         ch13_p2(o)=str2double(char(Potencias(75:77)'));
%         o=o+1;
%     end

end %fin del bucle de rellenar muestra
% plot(i,y);
end
a=1;b=1;c=1;d=1;e=1;f=1;g=1;h=1;k=1;l=1;m=1;p=1;o=1;

for j=1:22
    for i=1:n
        if(VECTOR_P1(i,j,theta)==0)

VECTOR_P1(i,j,theta)=mean(VECTOR_P1(find((VECTOR_P1(:,j,theta))<0),j,t
heta));

                end
                if(VECTOR_P2(i,j,theta)==0)

VECTOR_P2(i,j,theta)=mean(VECTOR_P2(find((VECTOR_P2(:,j,theta))<0),j,t
heta));

                end
            end
        end

        toc
        %copia de las medidas originales
        VECTOR_P1_raw(:, :, theta)=VECTOR_P1(:, :, theta);
        VECTOR_P2_raw(:, :, theta)=VECTOR_P2(:, :, theta);
        %%%%%%%%%%%%%%% llamamos para procesar los datos
        %%%%%%%%%%%%%%%
        processdata; %procesado preliminar para borrar datos erroneos
        %%representacion; %representación en tiempo real de la toma de
muestras (comentado para ahorrar cómputo)
        posp1=posp1+2;
        posp2=posp2+2;

        theta=theta+1;

end %while

% wait(0);
save('prueba_09032020.mat');

fprintf(mesa,'LD 1 DV');           % seleccionamos la mesa giratoria
fprintf(mesa,['LD' num2str(angulo_inicio) ' DG NP GO'] );

fclose(mesa);

```

## C. Procesado de muestras de la cámara anecoica (Process\_data.m)

```
load('medidas_rssi_09032020.mat')
vector_modas_p1=zeros(91,22);
vector_modas_p2=zeros(91,22);
vector_modas_p1_11=zeros(91,11);
vector_modas_p2_11=zeros(91,11);
vector_p1_raw_aux=VECTOR_P1_raw;
vector_p2_raw_aux=VECTOR_P2_raw;
vector_modas_p1_11_norm=zeros(91,11);
vector_modas_p2_11_norm=zeros(91,11);
lim=3; %limite para eliminar picos que se separen de los valores a su
lado

%eliminamos los valores imposibles (por encima de 35db)
for column=1:22
    for theta=1:91
        vector_p1_raw_aux(find(VECTOR_P1_raw(:,column,theta)>=
40),column,theta)=mean(VECTOR_P1_raw(find(VECTOR_P1_raw(:,column,theta)
)<=40),column,theta));
        vector_p2_raw_aux(find(VECTOR_P2_raw(:,column,theta)>=
40),column,theta)=mean(VECTOR_P2_raw(find(VECTOR_P2_raw(:,column,theta)
)<=40),column,theta));

        end

end

figure
for u=1:91
plot(1:100,vector_p2_raw_aux(:, :, 20))
hold on
end

for i=1:22
    for j=1:91
        vector_modas_p1(j,i)=mean(vector_p1_raw_aux(:,i,j));
        vector_modas_p2(j,i)=mean(vector_p2_raw_aux(:,i,j));
    end
end

%SOLO LAS MODAS
figure
for i=1:11
%plot(1:91,vector_modas_p1(:,i))
plot(1:91,vector_modas_p2(:,i))
hold on

end
legend('ch1','ch2','ch3','ch4','ch5','ch6','ch7','ch8','ch9','ch10','c
h11')
%MODAS QUITANDO LOS PICOS
for i=1:22
    for j=2:90
```

```

        if ((vector_modas_p1(j,i)>vector_modas_p1(j+1,i)+lim) &&
(vector_modas_p1(j,i)>vector_modas_p1(j-1,i)+lim)); %si el valor esta
5db por encima de los que tiene a los lados
        vector_modas_p1(j,i)=(vector_modas_p1(j-1,i) +
vector_modas_p1(j+1,i))/2; %cambia ese valor por la media de los que
tiene a los lados
        end
        if ((vector_modas_p2(j,i)>vector_modas_p2(j+1,i)+lim) &&
(vector_modas_p2(j,i)>vector_modas_p2(j-1,i)+lim)); %si el valor esta
5db por encima de los que tiene a los lados
        vector_modas_p2(j,i)=(vector_modas_p2(j-1,i) +
vector_modas_p2(j+1,i))/2; %cambia ese valor por la media de los que
tiene a los lados
        end
    end

%eliminar picos hasta el angulo 15
    for j=1:15
        if(vector_modas_p1(j,i)>-53)
            vector_modas_p1(j,i)=vector_modas_p1(j,i)-5;
        end
    end
    for j=74:91
        if(vector_modas_p2(j,i)>-53)
            vector_modas_p2(j,i)=vector_modas_p2(j,i)-5;
        end
    end

        %vector_modas_p2(76,i)=vector_modas_p2(76,i)-3
end
%MODAS SIN PICOS Y HACIENDO LA MEDIA DE LAS DOS PASADAS
% cont=11
% for theta=1:91
%     for column =1:11
%
vector_modas_p1_11(theta,column)=(vector_modas_p1(theta,column)+vector
_modas_p1(theta,column +cont))/2;
%
vector_modas_p2_11(theta,column)=(vector_modas_p2(theta,column)+vector
_modas_p2(theta,column +cont))/2;
%
%     end
% end

figure
for i=1:11
plot(1:91,vector_modas_p2(:,i),'r')
hold on

end
hold off
legend('ch1','ch2','ch3','ch4','ch5','ch6','ch7','ch8','ch9','ch10','c
h11')

%correcciones de valores concretos
%puerto 1
vector_modas_p1(44,1)=vector_modas_p1(44,1)-4.5;
vector_modas_p1(45,1)=vector_modas_p1(45,1)-4.7;
vector_modas_p1(45,7)=vector_modas_p1(45,7)-1;

```

```

vector_modas_p1(46,7)=vector_modas_p1(46,7)-4;
vector_modas_p1(47,7)=vector_modas_p1(47,7)-4;
vector_modas_p1(45,6)=vector_modas_p1(45,6)-1;
vector_modas_p1(46,6)=vector_modas_p1(46,6)-2;
vector_modas_p1(47,6)=vector_modas_p1(47,6)-3;
vector_modas_p1(47,9)=vector_modas_p1(49,6)-1;
%puerto 2
vector_modas_p2(45,1)=vector_modas_p2(45,1)-5;
vector_modas_p2(44,1)=vector_modas_p2(44,1)-3;
vector_modas_p2(46,1)=vector_modas_p2(46,1)-1;
vector_modas_p2(47,1)=vector_modas_p2(47,1)-2;

    for i=1:11
        vector_modas_p1(:,i)=smooth(vector_modas_p1(:,i),8,'sgolay');
        vector_modas_p2(:,i)=smooth(vector_modas_p2(:,i),8,'sgolay');
    end
% plot(1:91,vector_modas_p1(:,1:11))
%NORMALIZAR
for theta=1:91
    for column=1:11
        vector_modas_p1_11_norm(theta,column)= -
max(vector_modas_p1(20:35,column))+vector_modas_p1(theta,column);
        vector_modas_p2_11_norm(theta,column)= -
max(vector_modas_p2(54:68,column))+vector_modas_p2(theta,column);
    end
end
figure
for i=1:11
plot(1:91,vector_modas_p1_11_norm(:,i),'r')
hold on

end
legend('ch1','ch2','ch3','ch4','ch5','ch6','ch7','ch8','ch9','ch10','ch11')

%correccion de valores normalizados
%puerto 1
vector_modas_p1_11_norm(18,10)=vector_modas_p1_11_norm(18,10)-1;
vector_modas_p1_11_norm(18,9)=vector_modas_p1_11_norm(18,9)-1;
vector_modas_p1_11_norm(18,8)=vector_modas_p1_11_norm(18,8)-0.5;
vector_modas_p1_11_norm(19,10)=vector_modas_p1_11_norm(19,10)-0.5;
vector_modas_p1_11_norm(19,9)=vector_modas_p1_11_norm(19,9)-0.5;
vector_modas_p1_11_norm(19,8)=vector_modas_p1_11_norm(19,8)-0.2;
%puerto 2
vector_modas_p2_11_norm(50,1)=vector_modas_p2_11_norm(50,1)-0.5;
vector_modas_p2_11_norm(51,1)=vector_modas_p2_11_norm(51,1)-0.9;
vector_modas_p2_11_norm(52,1)=vector_modas_p2_11_norm(52,1)-0.4;
vector_modas_p2_11_norm(53,1)=vector_modas_p2_11_norm(53,1)+0.7;
vector_modas_p2_11_norm(56,1)=vector_modas_p2_11_norm(56,1)+0.1;
vector_modas_p2_11_norm(52,1)=vector_modas_p2_11_norm(52,1)+0.2;
vector_modas_p2_11_norm(76,2)=vector_modas_p2_11_norm(76,2)-9.2;
vector_modas_p2_11_norm(77,2)=vector_modas_p2_11_norm(77,2)-9.7;
vector_modas_p2_11_norm(76,3)=vector_modas_p2_11_norm(76,3)-8;
vector_modas_p2_11_norm(77,3)=vector_modas_p2_11_norm(77,3)-6;
vector_modas_p2_11_norm(78,3)=vector_modas_p2_11_norm(78,3)-5;
vector_modas_p2_11_norm(75,1)=vector_modas_p2_11_norm(75,1)-4.1;
vector_modas_p2_11_norm(76,1)=vector_modas_p2_11_norm(76,1)-4.2;
vector_modas_p2_11_norm(44,1)=vector_modas_p2_11_norm(44,1)-3;
vector_modas_p2_11_norm(45,1)=vector_modas_p2_11_norm(45,1)-3;

```



```

for i=1:11

vector_modas_p1_11_norm(:,i)=smooth(vector_modas_p1_11_norm(:,i),8,'sg
olay');

vector_modas_p2_11_norm(:,i)=smooth(vector_modas_p2_11_norm(:,i),8,'sg
olay');
end

for theta=1:91
    for column=1:11
        vector_modas_p1_11_norm(theta,column)= -
max(vector_modas_p1_11_norm(20:35,column))+vector_modas_p1_11_norm(the
ta,column);
        vector_modas_p2_11_norm(theta,column)= -
max(vector_modas_p2_11_norm(54:68,column))+vector_modas_p2_11_norm(the
ta,column);
    end
end

vector_modas_p2_11_norm(49:53,1)=vector_modas_p2_11_norm(49:53,1)+0.6;
vector_modas_p2_11_norm(54:56,1)=vector_modas_p2_11_norm(54:56,1)+0.15
;
vector_modas_p2_11_norm(54,1)=vector_modas_p2_11_norm(54,1)+0.1;
vector_modas_p2_11_norm(54:55,2)=vector_modas_p2_11_norm(54:55,2)-0.1;
figure
ejex=-90:2:90;
plot(ejex,vector_modas_p1_11_norm(:,1),'r','LineWidth',2)
hold on
plot(ejex,vector_modas_p1_11_norm(:,2),'b')
hold on
plot(ejex,vector_modas_p1_11_norm(:,3),'b')
hold on
plot(ejex,vector_modas_p1_11_norm(:,4),'b')
hold on
plot(ejex,vector_modas_p1_11_norm(:,5),'b')
hold on
plot(ejex,vector_modas_p1_11_norm(:,6),'g','LineWidth',2)
hold on
plot(ejex,vector_modas_p1_11_norm(:,7),'b')
hold on
plot(ejex,vector_modas_p1_11_norm(:,8),'b')
hold on
plot(ejex,vector_modas_p1_11_norm(:,9),'b')
hold on
plot(ejex,vector_modas_p1_11_norm(:,10),'b')
hold on
plot(ejex,vector_modas_p1_11_norm(:,11),'y','LineWidth',2)
hold on
plot(ejex,vector_modas_p2_11_norm(:,1),'r--','LineWidth',2)
hold on
plot(ejex,vector_modas_p2_11_norm(:,2),'b')
hold on
plot(ejex,vector_modas_p2_11_norm(:,3),'b')
hold on
plot(ejex,vector_modas_p2_11_norm(:,4),'b')
hold on
plot(ejex,vector_modas_p2_11_norm(:,5),'b')
hold on
plot(ejex,vector_modas_p2_11_norm(:,6),'g--','LineWidth',2)

```

```

hold on
plot(ejex,vector_modas_p2_11_norm(:,7),'b')
hold on
plot(ejex,vector_modas_p2_11_norm(:,8),'b')
hold on
plot(ejex,vector_modas_p2_11_norm(:,9),'b')
hold on
plot(ejex,vector_modas_p2_11_norm(:,10),'b')
hold on
plot(ejex,vector_modas_p2_11_norm(:,11),'y--','LineWidth',2)
hold off

legend('ch1','ch2','ch3','ch4','ch5','ch6','ch7','ch8','ch9','ch10','c
h11')
figure
angulos=-90:2:90;
angulos_max1=zeros(1,11);
for i=1:11

angulos_max1(i)=find(vector_modas_p1_11_norm(:,i)==max(vector_modas_p1
_11_norm(:,i)))
    angulos_max1(i)=angulos(angulos_max1(i))
end
scatter(1:11,angulos_max1,65,'filled');
% grid on
% grid minor
hold on
%
figure
angulos_max2=zeros(1,11);
for i=1:11

angulos_max2(i)=find(vector_modas_p2_11_norm(:,i)==max(vector_modas_p2
_11_norm(:,i)));
    angulos_max2(i)=angulos(angulos_max2(i));
end
scatter(1:11,angulos_max2,65,'filled','r');
% grid on
% grid minor

```

## D. Automatización de salto de canal según parámetros de QoS. (autoFH.sh)

```
#!/bin/bash

start_time=$(date +%s)
thr='1' #inicializo througput
thresholdThr='0' #inicializo límite de throughput
lost='1' #inicializo bitrate para actualizar
mediumLost='58.5' #umbral regular
greatLost='65' #umbral bueno
thresholdLost='52' #inicializo límite de pérdidas bitrate=52Mbps

#FIJAMOS LOS 11 CANALES POSIBLES
for i in 2462 2457 2452 2447 2442 2437 2432 2427 2422 2417 2412
do
    echo '-----'

    while [ $lost -lt $thresholdLost ];
    do
        while [ $thresholdThr -lt $thr ];
        do

#OBTENEMOS CANAL ACTUAL
while :
do
    chan=$(iw dev | grep 'channel' |awk '{print $11;}')
    echo "We are in Channel $chan"
    break
done #while obtener canal

#OBTENEMOS PROPORCIÓN DE PÉRDIDAS Y THROUGHPUT DEL CANAL

while :
do
    data=$(iw dev wlan1 station get d0:37:45:55:6e:a9) #dirección MAC del
receptor
    sent=$(echo $data | awk '{print $19;}')
    lostPackets=$(echo $data | awk '{print $25;}')
    echo "Number of sent packets: $sent"
    echo "Number of lost packets: $lost"
    lost=$(echo $data | awk '{print $41;}')
    echo

    #CÁLCULO DEL % DE PÉRDIDAS
    aux1='100'
    aux2=$lostPackets * $aux1
    percentage=$aux2 / $sent
    echo "Loss percentage: $percentage"

    #COMPROBACIÓN DE MCS
    if [ $lost -lt $mediumLost ]; then
        echo "Index 5 in MCS: regular communications"
```

```

else
    if [ $lost -gt $mediumLost ] && [ $lost -lt $greatLost]; then
        echo "Index 6 in MCS: good communications"
    else
        if [ $lost -gt $greatLost ]; then
            echo "Index 7 in MCS: perfect communications"
        fi
    fi
fi

thr_aux=$(echo $data | awk '{print $50;}')

#SOLUCIÓN A FALLO DE CAPTURA DE THROUGHPUT
bad="authenticated:"
if [ "$thr_aux" = "$bad" ]; then
    thr="NaN"
fi

if [ $thr_aux -ge $thr ];then #comparación >= máximo
    thr=$thr_aux
fi
#CÁLCULO DE 50% DE THROUGHPUT MÁXIMO
aux4='2'
thresholdThr=$thr / $aux4
echo "Throughput threshold: $thresholdThr"

break

done #while obtener pérdidas y rendimiento
done #while de threshold de throughput
done #while threshold de pérdidas
done #for canales

```

