Wi-Fi Insights: Explorando el Potencial del CSI en la Detección de Movimiento

Investigando un Modelo de Visión y Análisis del Entorno Mediante el Uso de Señales Wi-Fi

Horacio Manuel Lerma Andrés

Lydia Moragón Fandós

María Cristina Oliver Druet

IES Juan de Garay,
País Valencià

València, mayo 2025

Wi-Fi Insights: Explorando el Potencial del CSI en la Detección de Movimiento

Investigando un Modelo de Visión y Análisis del Entorno Mediante el Uso de Señales Wi-Fi

Horacio Manuel Lerma Andrés

Lydia Moragón Fandós

María Cristina Oliver Druet

Estudiante: Horacio Manuel Lerma Andrés
Bachillerato tecnológico, IES Juan de Garay
horlerand@alu.edu.gva.es

Profesoras: Lydia Moragón Fandós
Departamento de Matemáticas, IES Juan de Garay
l.moragon@edu.gva.es

María Cristina Oliver Druet
Departamento de Matemáticas, IES Juan de Garay
mc.oliver@edu.gva.es

IES Juan de Garay,
País Valencià

Trabajo de Investigación, 2023-2025

València, mayo 2025

Agradecimentos

A mis dos tutoras, Cristina y Lydia, por haberme animado a continuar y llevar a cabo este trabajo de investigación pese a las vastas dificultades y tardes enteras sin perder la esperanza. ¡Mil gracias!

Por supuesto, no me olvido del excelente y crucial papel de la profe de francés.

Resumen

El avance en los campos de la comunicación y la conectividad ha generado un importante impacto en la investigación de señales WiFi, especialmente con la evolución de la tecnología Wi-Fi y la introducción del estándar 802.11n (Wi-Fi 4), dando paso a los sistemas MIMO. En este trabajo, nos enfocaremos en la utilización de información del CSI para el estudio del entorno, específicamente en la detección de movimientos a partir exclusivamente de señales WiFi como entrada de datos. Esto nos permitirá realizar análisis incluso en entornos con obstáculos o con escasa iluminación. Además, exploraremos cómo esta tecnología puede salvaguardar el derecho a la privacidad en comparación con los sistemas de reconocimiento basados en luz, como los sistemas LiDAR.

Trataremos de implementar y valorar el potencial de un sistema compuesto por dos equipos interconectados a través de un punto de acceso (AP). Estos dos equipos actuarán como transmisor y receptor. El transmisor enviará un mensaje que el receptor reconocerá y utilizará para calcular el CSI. Una vez recopilada esta información, emplearemos MATLAB con el fin de representar gráficamente la amplitud y la fase de las señales capturadas.

Además, evaluaremos la viabilidad de desarrollar un sistema de este tipo utilizando recursos más accesibles y de bajo costo, lo que implicaría explorar alternativas a nuestro alcance. Este enfoque presenta desafíos importantes, entre ellos la precisión en la captura del CSI con equipos no especializados, la estabilidad de la conexión entre los dispositivos, así como las limitaciones en la capacidad de procesamiento y sincronización de los datos.

Keywords: 802.11n, Wi-Fi, Channel State Information (CSI), Tecnología MIMO (Multiple Input Multiple Output), Atheros CSI Tool, Señales inalámbricas, Movimiento, Detección de movimiento, Análisis de señales.

Índice de contenidos

Índice de figuras IX

Índice de tablas XI

Glosario XI

Siglas XII

1. Introducción **1**

2. Antecedentes **7**

2.1. Fundamentos teóricos 7

2.2. Punto de partida. Análisis de *DensePose From WiFi* [7] 8

3. Hipótesis **10**

4. Objetivos **12**

5. Materiales **14**

5.1. Materiales utilizados 14

5.1.1. Hardware 14

5.1.2. Software 15

5.1.3. Relación entre los materiales 16

5.2. Materiales descartados 17

5.2.1. Hardware 17

5.2.2. Software 19

6. Metología **22**

6.1. Búsqueda de material y dificultades 22

6.2. Diseño y ensamblaje del sistema 24

6.2.1. Laboratorio de pruebas para el estudio del CSI con la herramienta Atheros 24

6.2.2. Adaptación y diseño de la réplica de DensePose From WiFi . . . 25

6.3. Instalación, configuración y experimentación 26

7. Resultados **27**

7.1. Instalación del Sistema Operativo en los distintos dispositivos 27

7.1.1. Compilacón del kernel 28
7.1.2. Error con la compilación del nuevo kernel 29
7.2. Atheros CSI Tool 30
7.2.1. Comunicación entre el Tx y el Rx 30
7.2.2. Instalación Atheros CSI Tool 31
7.2.3. Preparación de Scripts de transmisión y recepción 32
7.2.4. Paquete de sondeo HT. Mensaje pactado 34
7.3. MATLAB 39
7.3.1. Scripts para MATLAB de la suite de *Atheros CSI Tool* 39
7.3.2. Representación gráfica de los datos obtenidos 40

8. Discusión y crítica de los aspectos éticos y morales **43**

9. Conclusión **46**

Bibliografía 49

Apéndices

A. Apéndice I **58**
A.1. Conceptos teóricos 58
A.2. Cálculo del CSI 59
A.2.1. Propagación por trazado de rayos 60
A.2.2. Modelo de Dispersión (*Scattering Model*) 61

Índice de figuras

1.1. Entorno de habitaciones conectadas por un pasillo. 2
1.2. Ejemplo exagerado para enfatizar la diferencia entre el mensaje original y recibido en la comunicación Wi-Fi. 3
1.3. El efecto de la presencia humana en la propagación de las señales Wi-Fi. . . 4
1.4. Entorno de habitaciones interconectadas creado por un punto de acceso. . 5

2.1. Correspondencia densa entre imágenes y un modelo de superficie 3D. Ejemplo extraído del paper DensePose [23]. 9

5.1. Panel de configuración del router LiveBox Fibra. 17
5.2. Segmentación de red para la captura y análisis del CSI. 17
5.3. Interfaces de red utilizadas con la NIC interna del portátil y el adaptador Wi-Fi USB. 18

6.1. GiGames motherboard. 23
6.2. Tarjetas Wi-Fi testeadas con el software Atheros CSI Tool. 23
6.3. Raspberry Pi 3B empleada en primera instancia como AP. 24
6.4. Disposición de los diferentes equipos para el estudio del CSI con la herramienta Atheros. 25
6.5. Aprendizaje transferido de una red maestra basada en imágenes a la red basada en WiFi, WiFi-DensePose RCNN. Foto extraída del paper DensePose From Wi-Fi. 26

7.1. Clásico panel de instalación de Ubuntu. 28
7.2. Simplificación de como el kernel conecta a las aplicaciones (capa de software) con la capa física (hardware) de un ordenador. 28
7.3. Compilación exitosa del kernel 4.1.10+ 30
7.4. Disposiciones y esquema de roles de cada equipo durante la captura del CSI. 31
7.5. Listado de archivos del repo Atheros-CSI-Tool-UserSpace-APP. 32
7.6. Salida del comando `recv_csi` al recibir los paquetes HT del emisor. 32
7.7. Salida del comando `send_Data`. 33
7.8. Campos de información para la celda {1,1}. 37
7.9. Monitor mostrando el tráfico de red capturado por Wireshark junto a un Baby Yoda con una antena Wi-Fi omnidireccional SMA-M. 38

7.10. Captura de Wireshark de los paquetes de sondeo interceptados enviados desde el emisor hasta el receptor. 38
7.11. Valores de CSI en función del tiempo. 40
7.12. Valores de RSSI de cada antena en función del tiempo. 41

A.1. Representación gráfica de la parte compleja y real del CSI. Imagen extraída de la web Wands del proyecto Atheros CSI Tool [9]. 62

Índice de tablas

5.1. Materiales y programas empleados y descartados durante el desarrollo del TdI. 15
5.2. Comparativa entre las diferentes herramientas para la captura de datos del CSI. 19

7.1. Información del estado capturada y almacenada a partir de cada Trama. . . 36

Glosario

Canal Medio físico o virtual por el que se transmite una señal. En comunicaciones inalámbricas, el canal incluye los efectos del entorno que afectan a la señal, como la atenuación, el ruido y el desvanecimiento. (*p. 8, 12, 15, 19, 20, 34, 36, 37, 39*)

MATLAB Entorno de programación y lenguaje especializado en cálculo numérico, análisis de datos y visualización. Ampliamente utilizado en ingeniería y ciencias para simulaciones, procesamiento de señales y modelado de sistemas. (*p. 30, 37, 39*)

MIMO Tecnología que emplea múltiples antenas en el transmisor y el receptor para mejorar la capacidad y fiabilidad de las comunicaciones inalámbricas. Permite transmitir varias señales simultáneamente a través del mismo canal. (*p. 18*)

Python Lenguaje de programación de alto nivel conocido por su sintaxis clara y su amplia aplicación en ciencia de datos, inteligencia artificial, desarrollo web y automatización. Popular en investigación y prototipado rápido. (*p. 39*)

RSSI Indicador de la potencia recibida de una señal inalámbrica. Se utiliza para estimar la calidad de la conexión, aunque no proporciona información tan detallada como el CSI. (*p. x, 36, 41*)

Señal piloto (Tx) Secuencia conocida enviada junto con los datos en un sistema de comunicaciones para facilitar la estimación del canal y la corrección de errores. Es esencial para recuperar la señal original tras atravesar el canal. (*p. 34*)

Trama Unidad básica de datos transmitida en redes de comunicación. En Wi-Fi, las tramas contienen campos que indican origen, destino, tipo de datos y control de errores. También se le suele denominar como "paquete de datos". (*p. xi, 31, 34–36, 38*)

Wireshark Herramienta de análisis de protocolos de red que permite capturar y examinar paquetes en tiempo real. Utilizada para diagnóstico de redes, seguridad informática e investigación de comunicaciones inalámbricas. (*p. ix, x, 37, 38*)

Siglas

AP Access Point. (*p. iv, 4, 14, 22*)

BIOS Basic Input/Output System. (*p. 29*)

CPU Central Processing Unit. (*p. 3, 5, 16*)
CSI Channel State Information. (*p. iv, 1–3, 5, 6, 8, 11, 12, 14, 15, 17–19, 24–26, 29–34, 36, 37, 39, 40*)

GPU Graphic Processing Unit. (*p. 3*)
GRUB GRand Unified Bootloade. (*p. 29*)

IEEE Instituto de Ingenieros Eléctricos y Electrónicos. (*p. 7*)
IoT Internet Of Things. (*p. 5, 17*)
IP Internet Protocol. (*p. 4*)

MIMO Multiple Input Multiple Output. (*p. iv, 20, 34*)

NIC Network Interface Card. (*p. 4, 14, 15, 18, 19, 22, 23*)

OEM Ondas Electromagnéicas. (*p. 5, 7*)
OFDM Orthogonal Frequency-Division Multiplexing. (*p. 34*)
OS Sistema Operativo. (*p. 27*)

PCI-E Peripheral Component Interconnect Express. (*p. 14*)

RAM Random Access Memory. (*p. 3, 14, 15, 19*)
RCNN Region Based Convolutional Neural Networks. (*p. 11*)
ROM Read Only Memory. (*p. 14, 15, 19*)
RSSI Received Signal Strength Indicator. (*p. 36*)

1 Introducción

Actualmente, el desarrollo de herramientas de detección de objetos y personas ha transformado nuestra vida cotidiana. Estas herramientas van desde sistemas de GPS en vehículos o teléfonos móviles, hasta sofisticados métodos de seguridad o de reconocimiento facial [2]. Estos avances tecnológicos han facilitado la monitorización y la recopilación de datos en una amplia variedad de contextos, lo que nos lleva a una concienciación considerable en materias de privacidad, especialmente en espacios públicos, donde se han instalado sistemas de vigilancia bajo la justificación de proteger al ciudadano [3].

Cada vez más, los individuos se enfrentan al desafío de proteger sus datos personales en un contexto donde la recopilación y el análisis de información son prácticas muy extendidas en nuestra sociedad. En este sentido, los métodos tradicionales de detección [4] que operan con luz, como cámaras y sistemas LiDAR [5], plantean preocupaciones debido a que recolectan una cantidad ingente de datos sensibles para el fin al que están destinados [6], que, en malas manos, podrían usarse contra nosotros. Además, cabe añadir que suelen ser costosos económicamente y requieren de una fuente de luz para poder operar.

Ante este desafío, nos surge la siguiente pregunta: ¿es posible desarrollar sistemas de detección que preserven la privacidad del individuo, considerando las limitaciones y preocupaciones asociadas? En este contexto, está más que justificado estudiar la posibilidad de utilizar otro tipo de herramientas para conservar el derecho a la intimidad. Estudios recientes [7] [8], sugieren que la tecnología Wi-Fi podría ser un campo prometedor para futuras investigaciones, ya que es menos invasiva y más accesible en comparación con los métodos convencionales. Asimismo, la presencia actual de la tecnología Wi-Fi en nuestra sociedad, la convierte en una opción accesible casi en cualquier entorno.

Estos estudios exploran el potencial de las señales Wi-Fi para obtener información relevante. Por ejemplo, en uno de los estudios se demostró que es posible predecir la pose humana utilizando únicamente señales Wi-Fi [7]. A su vez, otro estudio combinó dispositivos de seguimiento de salud personal con datos del Channel State Information

(CSI) para monitorear signos vitales y rastrear movimientos [8]. En este último caso, se obtuvieron datos como la frecuencia cardíaca y respiratoria.
Ante este paradigma, se plantea la siguiente pregunta: ¿de qué manera es posible que las señales Wi-Fi puedan ser utilizadas para la detección de movimiento, el reconocimiento de gestos e, incluso, la estimación de la pose humana? La respuesta radica en la forma en que estas señales interactúan con el entorno. A medida que las ondas Wi-Fi viajan por un espacio determinado, se ven sometidas a las perturbaciones causadas por los objetos y seres vivos presentes en dicho espacio, reflejándose en las paredes, atenuándose con la distancia o dispersándose al atravesar determinados materiales.
En el caso de que fuese posible capturar y analizar dichas variaciones con suficiente precisión, se podría extraer información valiosa sobre el entorno sin necesidad de recurrir a cámaras o sensores de carácter invasivo. Es en este punto donde se hace evidente la relevancia del Channel State Information, un conjunto de parámetros que permiten describir cómo la señal Wi-Fi se altera debido a su interacción con el medio.
Para una mejor comprensión del concepto, se plantea un ejemplo intuitivo que permite visualizar la propagación y el impacto del entorno sobre las señales y se irá escalando hacia un plano más teórico. Así pues, imagínese dos habitaciones conectadas por un pasillo de poco más de tres metros, así como en la figura 1.1. Ahora, colóquese en una de las habitaciones y, en la otra, ponga a un amigo. Si quieren comunicarse, no es necesario que se encuentren en la misma habitación; simplemente pueden hacerlo elevando la voz un poco más de lo normal. Esto es posible por el hecho de que el sonido necesita un medio material por el que viajar que, en este caso, es el aire, el suelo, las paredes o cualquier otro cuerpo que coloquemos entre medias —para simplificar la explicación, consideraremos las habitaciones inalterables ante lo que ocurra en el exterior y sin obstáculos en su interior—.

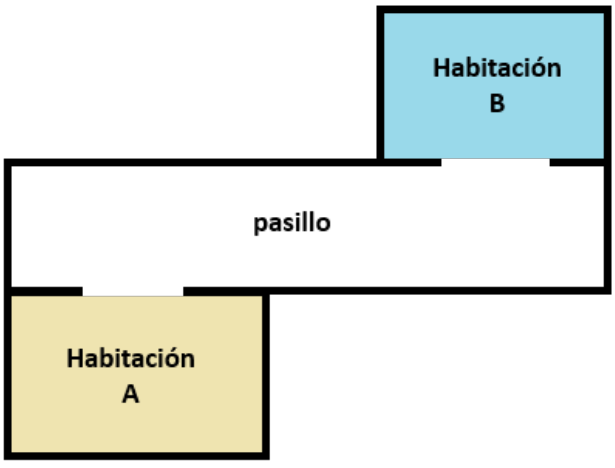


Figura 1.1: *Entorno de habitaciones conectadas por un pasillo.*

Si se ha observado que, en la práctica el Wi-Fi se comporta de igual manera que el sonido y no como la luz. A menos que se coloquen una serie de espejos de manera conveniente para que reflejen la luz de una habitación a otra, no será capaz de ver a su

amigo, pero sí hablar con él. Esto es debido a que la luz —a menos que se mencione lo contrario, se habla de la luz dentro del espectro visible— es incapaz de atravesar cuerpos opacos como las paredes.
Haciendo hincapié de nuevo en que esto no es un obstáculo en las señales Wi-Fi. Ahora, vamos a coger el pasillo de tres metros que separa ambas habitaciones y le vamos a añadir 10 metros más, siendo pues su longitud 13 metros. En este caso, si trata de comunicarse con su amigo con el mismo nivel de voz, su amigo deberá hacer un esfuerzo para poder escucharle, pues el sonido, a medida que se propaga, va disminuyendo su intensidad. Lo que acaban de experimentar es sumamente importante. El mensaje transmitido en un principio no es el mismo al recibido finalmente, es decir, ha sufrido un cambio. Más allá de la comunicación en sí, vamos a centrarnos exclusivamente en el cambio, que es lo que realmente nos importa; es decir, no vamos a pararnos a estudiar cuál era el mensaje original y cuál el final, sino la diferencia entre ambos.

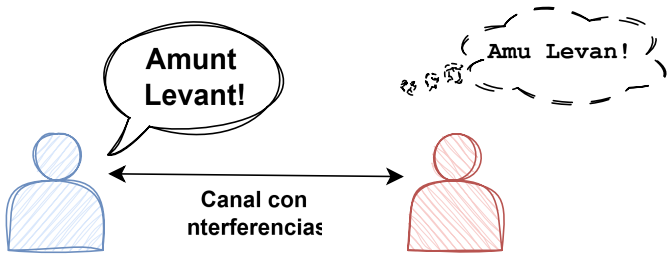


Figura 1.2: *Ejemplo exagerado para enfatizar la diferencia entre el mensaje original y recibido en la comunicación Wi-Fi.*

Recapitulemos, pero añadamos nuevos términos. Hemos planteado la situación de las dos habitaciones comunicadas por un pasillo. Este, es el entorno o medio. Su voz, que se transmite por el medio rebotando entre las paredes, es la señal, la cual se amortigua según avanza —sufre una atenuación— y llega con un pequeño retardo. Si su amigo tuviera un micrófono especial —como una tarjeta Wi-Fi modificada—, podría medir esas distorsiones para saber más sobre el "camino" que su voz ha recorrido ya que, al haber pactado un mensaje entre ambas partes, se conoce tanto el mensaje original "Amunt Levant!" como la versión a la cual ha tenido acceso el receptor, "Amu Levan!". Esto es, grosso modo, lo que estudia el CSI.
Ahora bien, como se ha mencionado, no nos sirve cualquier «micrófono», sino uno "especial". Antes de continuar con la búsqueda de este instrumento imprescindible, se hará un breve recorrido por los componentes que conforman un ordenador.
El procesador o CPU es el cerebro que da órdenes e instrucciones al resto de componentes y está formado por uno o varios núcleos que se encargan de realizar millones de operaciones por segundo. Los recursos necesarios para el funcionamiento del sistema se almacenan en la memoria dinámica RAM, y estática, HDD/SSD.
Otro componente esencial es la tarjeta gráfica o GPU, que se encarga de "traducir" las instrucciones de la CPU en imágenes comprensibles para el usuario. No obstante, es plausible que el lector haya escuchado que en los últimos años ha habido escasez de estas tarjetas, y esto se debe, en parte, a que también se usan para fines que no tienen

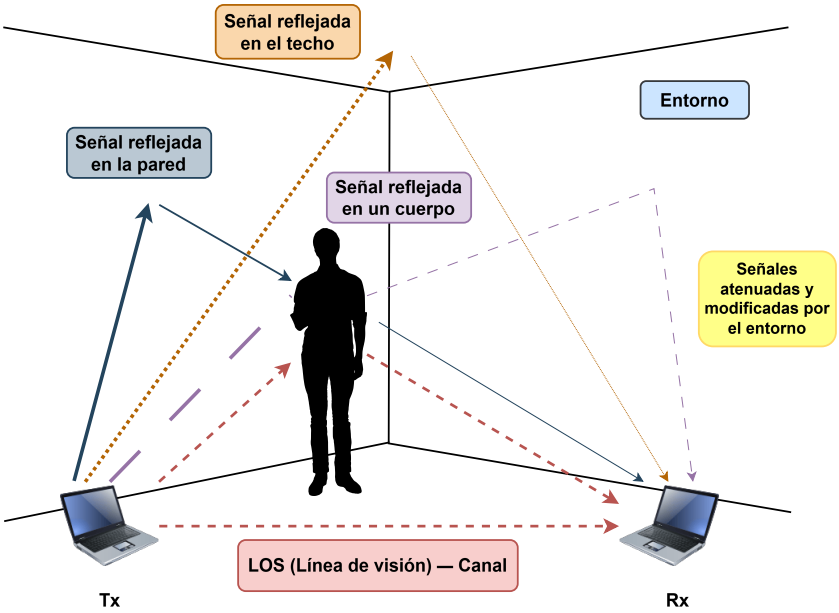


Figura 1.3: *El efecto de la presencia humana en la propagación de las señales Wi-Fi.*

que ver directamente con su uso convencional, como la minería de criptomonedas [21] o el entrenamiento de redes neuronales [22]. Esto ocurre porque las tarjetas gráficas tienen una gran capacidad de procesamiento, lo que las hace útiles para realizar cálculos complejos, más allá de su función tradicional que es generar imágenes a través de matrices de píxeles.

La NIC, o Tarjeta de Interfaz de Red (en inglés, *Network Interface Card*), es un componente presente en la mayoría de los ordenadores. Su función principal es permitir que estos se conecten con otros equipos sin necesidad de cables. Para conseguirlo, recibe (Rx) las señales analógicas que viajan a través del aire en las bandas de frecuencia de 2.4 GHz y 5 GHz, y las transforma en datos que el ordenador puede entender —paquetitos de unos y ceros—. Además, también puede hacer el proceso inverso: transmitir información (Tx) en la misma banda, convirtiendo los paquetes de 0s y 1s en señales analógicas que se difractan, reflejan y atenúan al propagarse a través del medio. Gracias a estas tarjetas, dos equipos pueden comunicarse entre sí sin cables. Sin embargo, para que esto funcione correctamente, debemos configurarlas para que no solo se comuniquen con el router —para acceder a internet—, sino también directamente entre ellas, como si fueran esas personas que hablan entre habitaciones.

En este ejemplo de las dos habitaciones es fácil imaginarnos a la NIC como un "micrófono especial" que capta las señales Wi-Fi del otro dispositivo. Para que la comunicación se dé de forma ordenada y efectiva, interviene otro elemento importante: el punto de acceso (AP). Este actúa como una especie de organizador, asignando a cada equipo una dirección virtual (IP), como si se tratara del número de habitación de cada uno, y asegurándose de que ambos estén "en el mismo edificio", es decir, en la misma red.

Durante esta comunicación, una de las tarjetas transmitirá una serie de paquetes de datos con información específica: la hora exacta del envío, algún mensaje —como un

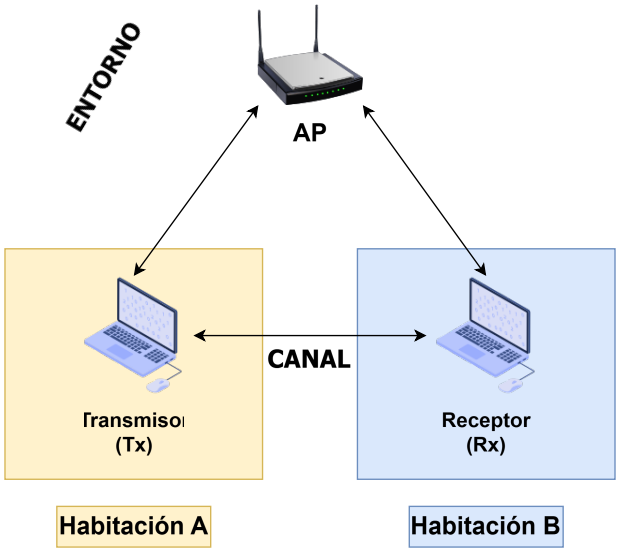


Figura 1.4: *Entorno de habitaciones interconectadas creado por un punto de acceso.*

saludo—, la potencia con la que se transmite, el tamaño de los datos, entre otros detalles. La otra tarjeta, que de antemano sabe que mensaje ha enviado el otro equipo, capta estos paquetes, los transforma en un formato que el procesador o CPU pueda interpretarlos y, por último, realiza las operaciones necesarias para comparar la señal recibida con la original extrayendo de esta comparación la información útil. Esto es, el CSI.

En este escenario, nos planteamos la cuestión de determinar el número de individuos en una habitación, así como la pose —si están sentados, de pie, haciendo el pino— o la presencia de movimiento y su cantidad. Eliminando las restricciones impuestas por las condiciones lumínicas y los obstáculos. No obstante, la relevancia de este estudio radica en la capacidad de establecer una relación entre los datos procesados y cualquier otra situación del entorno inferida por el comportamiento natural de las OEM (Ondas Electromagnéticas). Esta, es la idea principal del presente trabajo: aprovechar cómo se comportan naturalmente estas señales para inferir lo que está ocurriendo a nuestro alrededor y aplicarlo en aspectos como la domótica con el uso de dispositivos IoT (del inglés, *Internet of Things*). La segunda tarjeta se encarga de recibir dichos paquetes —previamente conocidos—, transformarlos para que la CPU pueda interpretarlos y, en última instancia, llevar a cabo las operaciones necesarias para extraer información relevante del proceso de comunicación al comparar ambas señales.

Esta investigación tiene un valor personal significativo para mí, ya que me da la oportunidad de explorar materias que me apasionan como son las telecomunicaciones y la informática. En el instituto no suelo tener la ocasión de profundizar en estos temas, así que este estudio es algo más que un simple trabajo. Desde el principio de la investigación, mi meta es profundizar todo lo posible en estos campos y aprender de la experiencia de investigar en su conjunto. Además, considero que trata temas, a mi parecer, actuales, que siguen siendo objeto de investigación y estudio, como es el análisis

del CSI para obtener datos que pueden llegar a arrojar información valiosa si se les aplica un tratamiento adecuado.
Con todo esto y persiguiendo el objetivo de estudiar las propiedades del CSI en el entorno, nos planteamos el siguiente trabajo de investigación en el que, tras mostrar unos antecedentes para conocer los fundamentos teóricos (2) en los que se basa este estudio, planteamos unas hipótesis (3) y unos objetivos (4), así como los materiales (5) y la metodología (6) empleados para llegar a los resultados obtenidos (7) y establecer nuestras propias conclusiones (9).

2
Antecedentes

Para adentrarnos en el contexto general de esta investigación, resulta esencial comprender ciertos conceptos fundamentales, tales como las propiedades asociadas a las ondas. En los siguientes apartados, trataremos los puntos esenciales para favorecer la comprensión del trabajo. Aquellos conceptos menos significativos figuran en el apéndice A.

2.1. Fundamentos teóricos

Wi-Fi

El Wi-Fi es una tecnología de comunicación inalámbrica que permite la conexión entre dispositivos electrónicos mediante Ondas Electromagnéicas (OEM). Basada en los estándares de la familia IEEE 802.11, opera principalmente en las bandas de frecuencia de los 2.4 GHz y 5 GHz. Esta tecnología es ampliamente utilizada en hogares y empresas para proporcionar acceso a Internet y facilitar la interconexión entre dispositivos.

IEEE 802.11

Es la forma técnica de llamar a la tecnología Wi-Fi y se refiere a la familia de normas inalámbricas creada por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE). En este documento haremos referencia principalmente a dos estándares:

- 802.11n: Wi-Fi 4
- 802.11ac: Wi-Fi 5

CSI (Channel State Information)

En el ámbito de las comunicaciones inalámbricas, la información del estado del canal (del inglés, *Channel State Information*) constituye un componente esencial que permite la comprensión de las características y propiedades inherentes a un enlace de comunicación. Este conjunto de datos describe el proceso de propagación de la señal Wi-Fi

desde el transmisor hasta el receptor, proporcionando una representación cuantificable de los efectos físicos asociados a fenómenos como la dispersión, el desvanecimiento y la atenuación de la potencia de la señal con la distancia. En la praxis, el objeto de estudio son las variaciones experimentadas por la información durante su transmisión en un entorno específico. En capítulos subsecuentes se abordará con mayor profundidad este concepto, el cual constituye la justificación fundamental de la presente investigación.

Raw signals (Señales en crudo)

Las señales en crudo se refieren a los datos sin procesar que se recopilan directamente de un dispositivo o sistema de medición. Estas señales no han sido sometidas a ningún tipo de filtrado, transformación o análisis, y representan la información en su forma más básica y original. Por lo general, la información que nos proporcionan no es legible a simple vista hasta que se le aplica un tratamiento adecuado.

Cálculo del CSI

El cálculo del CSI es el proceso mediante el cual se determina la respuesta del Canal de comunicación. Esta respuesta incluye información acerca de dónde está el cliente, a qué distancia se encuentra, etc. Dependiendo del entorno donde se propaguen las señales Wi-Fi, existen diferentes modelos para describir dicha propagación. De nuevo, en el apéndice A explicamos los dos modelos que podemos encontrar.

2.2. Punto de partida. Análisis de *DensePose From WiFi* [7]

El estudio en cuestión propone una alternativa a los métodos convencionales de estimación de la pose, tales como las cámaras RGB, las tecnologías LiDAR y radar, que presentan restricciones ocasionadas por problemas de oclusión, iluminación y privacidad. Además, estos métodos son costosos y requieren un alto consumo de energía. En contraste, las señales Wi-Fi ofrecen una alternativa más asequible y accesible, al tiempo que preservan la privacidad al no capturar imágenes visuales. Tomando como premisa la tecnología Wi-Fi, el propósito de los autores consiste en desarrollar un método para estimar la pose densa del cuerpo humano utilizando únicamente señales Wi-Fi. Este objetivo se logra mediante el uso de redes neuronales profundas que mapean la fase y la amplitud de las señales Wi-Fi a coordenadas UV dentro de 24 regiones del cuerpo humano. Las coordenadas UV representan un sistema de coordenadas bidimensional utilizado para mapear una superficie 3D a un plano 2D
La estimación de la pose densa humana se basa en el mapeo de todos los píxeles de una imagen RGB 2D correspondiente con un cuerpo humano y su plasmación sobre una superficie 3D que recrea el cuerpo humano. El término “pose densa” hace referencia a otro estudio: *DensePose* [23], que describe una técnica de visión por computadora que permite estimar una correspondencia densa entre una imagen de una persona y un modelo 3D del cuerpo humano, como se puede apreciar en la figura 2.1. A diferencia

de los métodos tradicionales, que sólo detectan puntos clave como codos y rodillas, *DensePose* proporciona una correspondencia detallada para cada píxel de la imagen con una ubicación específica en la superficie del cuerpo 3D. En este artículo del que partimos, *DensePose from WiFi*, se presenta la integración de este sistema de visión, pero introduciendo como datos de entrada señales Wi-Fi en lugar de píxeles, aprovechando la información de la amplitud y la fase para inferir la posición y movimiento del cuerpo, que, a su vez, es posible gracias a la reflexión y dispersión de las señales Wi-Fi.

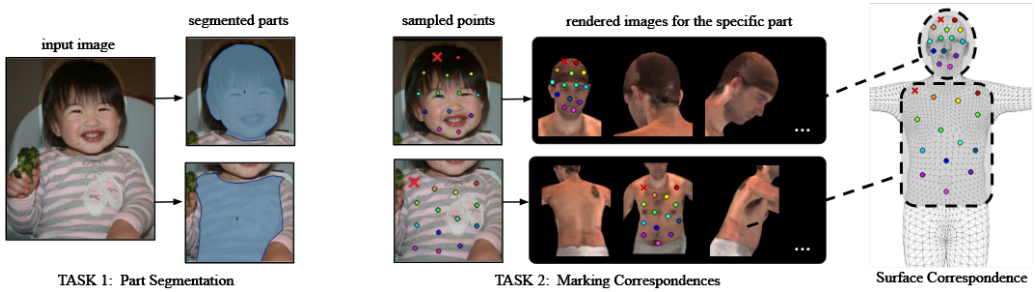


Figura 2.1: *Correspondencia densa entre imágenes y un modelo de superficie 3D. Ejemplo extraído del paper DensePose* [23].

3
Hipótesis

En este contexto, presentamos las siguientes hipótesis, las cuales abordan aspectos clave relacionados con la estabilidad de las señales, el impacto de dispositivos electrónicos externos, la relación entre los valores de amplitud y fase del CSI, el efecto del número de personas en un entorno analizado, y las posibles dificultades en la interpretación de los datos.

H.1. **Se puede diseñar un sistema de captura y análisis del CSI con materiales más sencillos, accesibles y de bajo coste.** A diferencia de otras investigaciones, buscamos utilizar un hardware disponible ya en muchos de los equipos del mercado y que sólo sea cuestión de modificar el software para poder operar con el CSI.

H.2. **En una habitación vacía el CSI se mantiene constante.** En un espacio aislado, es decir, libre de interferencias electrónicas y físicas que puedan interferir en la recopilación del CSI, los valores de la amplitud y la fase permanecerán dentro de un rango relativamente constante. Específicamente, se espera que la representación gráfica de estos valores en función del tiempo exhiba una tendencia estable o lineal, con cambios mínimos o pequeños saltos. Este comportamiento reflejaría la ausencia de perturbaciones relevantes en el entorno, pues el CSI estudia los cambios que experimenta la señal al propagarse por un medio, pero al no haber perturbaciones en él, la señal recibida será prácticamente idéntica a la original lo que permitiría una adquisición más precisa y consistente de los datos del CSI.

H.3. **Los dispositivos electrónicos interfieren en la captura de los datos.** Los dispositivos electrónicos que emiten en la misma frecuencia que los estándares WiFi 802.11n y 802.11ac, es decir, dentro del espectro de los 2.4GHz y 5GHz, afectarán significativamente a los datos recopilados, generando ruido electrónico, lo que puede llegar a dificultar el análisis de las señales.

H.4. **El aumento del número de personas dentro de un entorno analizado aumenta el ruido en los datos recolectados.** Como hemos explicado, las personas actúan como cuerpos que reflejan y refractan las señales Wi-Fi, modificando así la línea de visión.

H.5. **Es posible predecir patrones de movimiento en un entorno a partir de las métricas del CSI.** Al obtener una relación entre los valores de la amplitud y la fase del CSI para predecir el movimiento en un espacio, esperamos descubrir nuevas relaciones que nos arrojen información valiosa.

H.6. **La capacidad de trabajar con múltiples personas en un mismo espacio dependerá del hardware utilizado para estudiar el CSI.** En la investigación *DensePose From WiFi* se reportan errores al estudiar a tres o más sujetos en la misma habitación —también podría ser un error de entrenamiento de la RCNN, como indican en el estudio—.

4
Objetivos

El propósito fundamental de esta investigación es explorar el potencial del CSI de las señales Wi-Fi en la detección de movimiento. En el ámbito de la recopilación de datos de las ondas Wi-Fi, se destaca la relevancia del CSI, que, como hemos descrito, proporciona información relacionada con el Canal de comunicación. Inicialmente, estos datos pueden parecer carentes de sentido hasta que se les aplica un tratamiento adecuado. Se podría conceptualizar el CSI como una caja negra compuesta por múltiples componentes aparentemente desordenados. No obstante, al disponerlos de manera ordenada, estos datos se vuelven altamente útiles para el análisis de aspectos específicos del entorno, como se evidencia en el artículo de DensePose from Wi-Fi con la estimación de la pose humana. En consecuencia, el propósito de este estudio es identificar patrones emergentes que permitan una organización eficiente de los datos, con el fin de generar la información requerida.
Una vez cumplido este objetivo, exploraremos nuevos marcos teóricos centrándonos en aspectos más concretos del movimiento.
A partir del objetivo principal se plantean diferentes objetivos secundarios orientados a evaluar la aplicabilidad práctica del sistema en distintos contextos y bajo condiciones tecnológicas accesibles como son:

- Evaluar la posibilidad de implementar este sistema para la detección de caídas en personas con movilidad reducida o accidentes domésticos.
- Explorar la aplicación de este sistema en una amplia variedad de implementaciones y automatizaciones domóticas.
- Realizar el estudio con hardware accesible y barato.

Por último, aunque anticipamos dificultades en la interpretación de la información de las señales debido a la complejidad de los conceptos matemáticos y la especificidad del ámbito de las telecomunicaciones, confiamos en poder superar estos desafíos a medida que avancemos en el estudio.
Lo que nos llevó a empezar a investigar y proponernos llevar el trabajo a un marco más formal y poder llegar a presentarlo, fue un estudio realizado por unos ingenieros de

la universidad de Pittsburgh, *DensePose from Wi-Fi* . Dicho trabajo, ya presentando en Antecedentes 2.2, intuíamos la posibilidad de desarrollar un sistema que, mediante el análisis de las señales Wi-Fi, sea capaz de predecir e inferir comportamientos relacionados con el movimiento, tales como cambios bruscos en el estado de un individuo. Este enfoque podría resultar de gran utilidad para la detección de caídas en personas con movilidad reducida o accidentes domésticos, sin necesidad de instalar sensores en toda la vivienda, ya que la tecnología Wi-Fi sería suficiente para la recopilación de datos. Además, este mismo sistema podría emplearse para una amplia variedad de implementaciones y automatizaciones domóticas.

5 Materiales

En primera instancia, necesitamos establecer un sistema funcional para la recolección y análisis de los datos del CSI. Posteriormente, una vez que hayamos definido el propósito para el cual se necesitan los datos recolectados, procederemos con el diseño de un algoritmo destinado a extraer y transformar estos datos en información relevante. A continuación, detallaremos los materiales diferenciando entre hardware y software. La lista de estos componentes se encuentra en el cuadro 5.1.

5.1. Materiales utilizados

5.1.1. Hardware

Para la captura de los datos del CSI, utilizamos el siguiente hardware, seleccionado tras una serie de pruebas y evaluaciones para garantizar su compatibilidad y estabilidad:

- Un punto de acceso (AP). Optamos por el router doméstico debido a su fácil acceso, ya que se encuentra prácticamente en todos los hogares.
- Una placa base de una máquina tragaperras de la marca GiGames equipada con 4 GB de RAM y 40 GB de ROM.
- Un equipo portátil Toshiba Tecra S4-10C proporcionado por el instituto.
- Tarjeta de red Atheros AR9462 AR5B22 Mini PCI-E 802.11N[1]. Hemos seleccionado esta tarjeta de red como la opción principal para la captura y el análisis del CSI por ser una NIC Dual Band con soporte para dos antenas, por su extensa presencia en equipos, y por haber sido señalada como compatible por los desarrolladores de la herramienta de *Atheros CSI Tool*.

El hardware seleccionado es algo antiguo, lo que favorece la compatibilidad con la herramienta *Atheros CSI Tool*, de la que hablaremos más adelante, y permite modificaciones y ampliaciones. Además, esta elección contribuye a alcanzar el objetivo de emplear material económico y accesible.

[1] Tanto AR9462 como AR5B22 hacen referencia al mismo modelo de NIC.

Lista del hardware y software utilizados				
Hardware				
Cantidad	**Material**	**Sistema Operativo**	**Versión del kernel**	**Empleado**
1	**Toshiba Tecra S4-10C**	Ubuntu 14.04 LTS x64	4.1.10+	Sí
1	**Placa base GiGames**	Ubuntu 14.04 LTS x64	4.1.10+	Sí
2	Atheros NIC (AR9285) - dual band	x	x	No
2	**Atheros NIC (AR9462) - dual band**	x	x	Sí
1	**Atheros USB Adapter (AR5B22) - single band**	x	x	Sí
1	Access Point (Raspberry Pi 3B+/4)	Raspberry Pi OS Lite x64	6.6.20	No
1	**Livebox Fibra**	Arcadyan Firmware	x	Sí
1	**Placa de desarrollo Wi-Fi ESP32**	x	x	Sí
Software				
Nombre	**Función**	**Sistema Operativo**	**Versión**	**Empleado**
VMWare Workstation Pro	Creación de máquinas virtuales	Windows 10	x	Sí
Ubuntu	Sistema base para la captura del CSI	Ubuntu Desktop	14.04 LTS	Sí
Atheros CSI Tool	Captura de datos del CSI	Ubuntu 14.04	GitHub Repo	Sí
Linux 802.11n CSI Tool	Captura de datos del CSI	Desktop	GitHub Repo	No
Nexmon CSI Tool	Captura de datos del CSI	Android	GitHub Repo	No
ESP32 CSI Tool	Captuar de datos del CSI	x	GitHub Repo	No
Software Defined Radio (SDR)	Captura de datos del CSI	x	x	No
MATLAB Online	Análisis de datos del CSI	Navegador Web	Última versión	Sí
hostapd	Punto de acceso Wi-Fi	Raspberry Pi OS Lite x64	2.11	No
Arduino IDE	IDE para las placas ESP32	Arch	2.3.4-2	No
Python3 (server)	Transferencia de archivos	Ubuntu 14.04	3.x	Sí
Neovim	Editor de texto	Arch	x	Sí
Wireshark	Análisis del tráfico de red	Arch / Windows 10	4.2.0	Sí

Tabla 5.1: *Materiales y programas empleados y descartados durante el desarrollo del TdI.*

5.1.2. Software

En cuanto al software, se emplean los siguientes programas y versiones:

- **Sistema operativo.** Tanto para la placa base como para el portátil, se utiliza la versión 14.04 LTS de Ubuntu.
- **Kernel.** En cuanto al kernel, se emplea una versión modificada con drivers específicos para las NICs Wi-Fi basadas en la PCI-e de Atheros, como las de las series AR95xx, AR94xx y AR93xx [13]. Esta versión es la 4.1.10+ y está disponible como software de código abierto en GitHub [14] bajo la licencia GPL.
- **Suite de Atheros CSI Tool.** La herramienta utilizada para capturar y calcular el CSI es *Atheros CSI Tool*. Se emplea la suite de programas recogida en el repositorio *Atheros CSI Tool UserSpace APP* [15], que incluye los scripts necesarios para enviar y recibir datos, así como los archivos requeridos para su tratamiento en MATLAB.
- **Máquinas virtuales con diferentes versiones de Ubuntu.** Realizamos pruebas en máquinas virtuales (VMs) con diferentes entornos para identificar el que presentara menos complicaciones. Posteriormente, configuramos el sistema de análisis y estudio en máquinas físicas con las mismas características. Finalmente, creamos los entornos con 4 GB de RAM, 2 núcleos, 40 GB de ROM y arquitectura AMD de 64 bits y la versión de Ubuntu 14.04 LTS.
- **Atheros CSI Tool.** El software *Atheros CSI Tool* [9] está diseñado para extraer la Información del Estado del Canal en redes Wi-Fi basadas en el estándar IEEE 802.11n. La herramienta en cuestión presenta varias características que la convierten en una pieza clave para esta investigación:
    - Permite la extracción de CSI en tiempo real ofreciendo la posibilidad de una investigación más exhaustiva, al permitir el acceso a los datos en su estado original.

- Se ha diseñado específicamente para funcionar con tarjetas de red basadas en chipsets Atheros, como los modelos AR9580 y AR9285, lo que garantiza su compatibilidad con hardware accesible.
- Opera a través de una interfaz en espacio de usuario lo que simplifica su uso.
- Permite ser instalada en la mayoría de los equipos utilizados, desde ordenadores de uso doméstico hasta las CPUs de las tragaperras.

Cabe señalar que, para su correcto funcionamiento, requiere un sistema basado en Linux Ubuntu 14.04 LTS con una versión del kernel modificada (4.1.10+), ya que la versión estándar del sistema operativo no proporciona acceso a los datos CSI de la capa física.

5.1.3. Relación entre los materiales

En primer lugar, establecimos el entorno de trabajo, es decir, instalamos el software y el hardware necesarios en cada equipo para operar con los diferentes periféricos.

En ambos dispositivos, tanto en la placa base como en el ordenador portátil, instalamos la versión 14.04 de Ubuntu que viene por defecto con la versión del kernel 4.4.0-142-generic. Sin embargo, como hemos mencionado, a pesar de que las NICs comerciales trabajan con los datos del CSI, estos permanecen ocultos al usuario, por lo que debemos programarlas con el fin de liberar esta información. Para ello, compilamos una segunda versión para el kernel, la 4.1.10+ [14], que incluye el módulo con el driver ath9k modificado.

En el caso del router, optamos por dos vías: emplear una Raspberry Pi 3 B+ como AP o usar el router doméstico. Ambas opciones son igual de válidas.

Para el caso de la Raspberry Pi, la configuramos usando la última versión del software Raspberry Pi OS Lite (64-bit). Una vez instalado el SO (Sistema Operativo), configuramos el servicio *hostapd*, cuya función es levantar un punto de acceso.

En cuanto al router doméstico, lo ideal habría sido utilizar un router independiente del de la compañía para instalarle un firmware modificado y tener pleno control sobre él. Sin embargo, no disponíamos de ningún otro router a mano, por lo que con la opción de "Wi-Fi de Invitados" levantamos una subred paralela (192.168.0.1/24) a la red principal (192.168.1.1/24) donde únicamente se encontraban los equipos transmisor y receptor con sus respectivas NICs.

Para poder acceder a los datos recogidos por el receptor sin utilizar ningún medio físico como un USB, instalamos el adaptador Wi-Fi USB en el receptor, con el que montamos un servidor en Python dentro de la red principal y aislado de la subred.

Figura 5.1: *Panel de configuración del router LiveBox Fibra.*

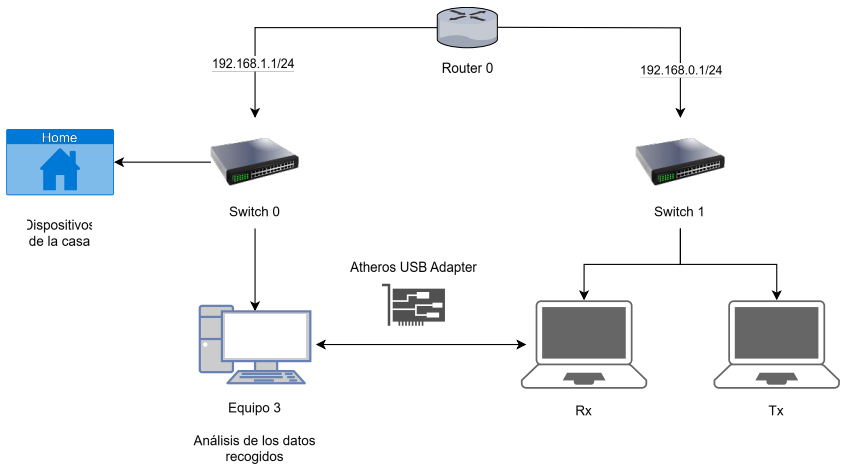


Figura 5.2: *Segmentación de red para la captura y análisis del CSI.*

5.2. Materiales descartados

5.2.1. Hardware

Durante la fase inicial del proyecto, probamos varios dispositivos que, aunque resultaron útiles para la experimentación, finalmente los descartamos debido a limitaciones técnicas, problemas de compatibilidad o resultados inconsistentes.

- **Placas de desarrollo Wi-Fi ESP32.** Consideramos estas placas por su popularidad en proyectos de IoT (*Internet of Things*) y su bajo costo [11]. Sin embargo, las descartamos debido a sus limitaciones para capturar y procesar el CSI de manera precisa y en tiempo real. No solo eso, algunas de ellas ni siquiera superaron la compilación del firmware correspondiente [10]: se calentaban, dejaban de funcionar y el proceso resultaba bastante complejo.
- **Placas base de diferentes máquinas que han dejado de usarse.** Reutilizamos placas base de máquinas antiguas para evaluar su capacidad en la captura del

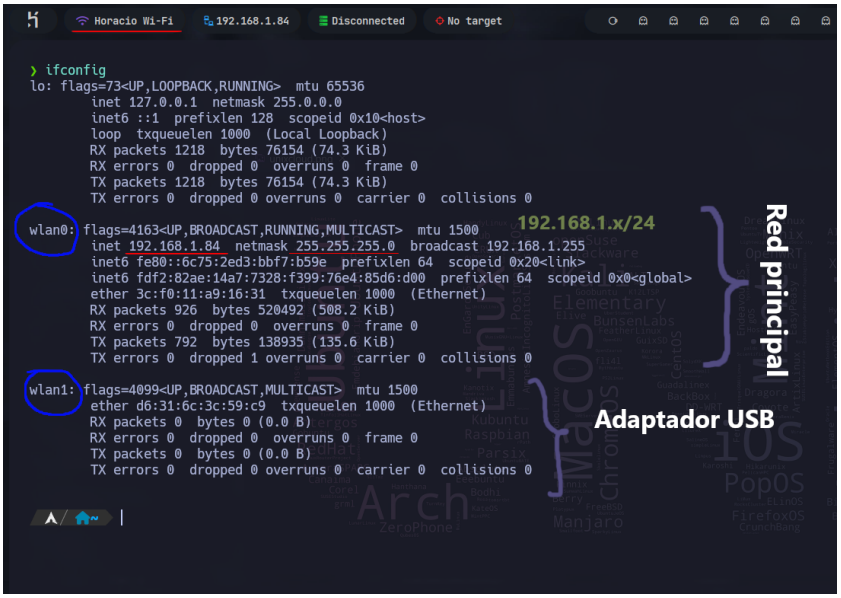


Figura 5.3: *Interfaces de red utilizadas con la NIC interna del portátil y el adaptador Wi-Fi USB.*

CSI. A pesar de que no pudimos utilizar estas placas base para nuestra investigación el motivo de descartarlas no fue por incompatibilidad sino por limitaciones técnicas.

- **Raspberry Pi 3B/B+ y Raspberry Pi 4.** Aunque las Raspberry Pi resultaron útiles como puntos de acceso, optamos por utilizar el router doméstico debido a que no era necesario usar dispositivos externos para cumplir la misma función.
- **Tarjetas de red.**

  - **Intel 5300 NIC.** Consideramos esta NIC por su soporte para la tecnología MIMO (*Multiple-Input & Multiple-Output*) con tres antenas, lo que ofrecería una capacidad de captura de datos más precisa. Sin embargo, tuvimos que descartarla porque las placas base disponibles en el proyecto solo soportaban la conexión de hasta dos antenas.
  - **Atheros AR9285.** Probamos este modelo porque venía integrado en las placas base y estaba presente en la lista de hardware técnicamente compatible con la herramienta *Atheros CSI Tool* [13], no obstante, tras las pruebas, resultaron incompatibles.
  - **TL-WDR4300.** Consideramos este router porque puede operar en la banda dual de 2.4GHz y 5GHz [12]. Finalmente, decidimos descartarlo porque estaba descatalogado y porque su tarjeta de red, integrada en la placa, limita la manipulación o sustitución de la NIC.
  - **Adaptador USB Atheros AR5B125.** Consideramos este dongle porque utiliza el mismo chip que el modelo AR9462, compatible con la herramienta *Atheros CSI Tool*. Durante las pruebas, no logramos capturar ni procesar correctamente el CSI, por lo que descartamos el adaptador.

Herramienta	Chipset Soportado	Estándar Wi-Fi	Características Clave	Razones para (no) usarla en el TdI
Linux 802.11n CSI Tool	Intel Wi-Fi 5300	802.11n	Captura CSI en 30 subportadoras, firmware modificado, herramientas en MATLAB	No compatible con el hardware disponible (NIC Intel 5300 requerida).
Atheros CSI Tool	Atheros AR9580, AR9285, etc. [13]	802.11n	Captura CSI en tiempo real, scripts en espacio de usuario, análisis en MATLAB/Python	Compatible con el hardware del proyecto, acceso directo a los datos CSI.
Nexmon CSI Tool	Broadcom BCM4339, BCM4366	802.11n/ac	Soporta 80 MHz, captura CSI en dispositivos móviles y routers	Requiere hardware no soportado por las placas base del proyecto.
ESP32 CSI Tool	ESP32 (Wi-Fi SoC)	802.11n	Captura CSI con microcontrolador, visualización en móvil, almacenamiento en microSD	Datos imprecisos, placas con sobrecalentamiento, dificultad en compilación.
Software Defined Radio	Varios SDR (HackRF, RTL-SDR)	Variable	Altamente configurable, permite analizar cualquier señal	Alto costo y requerimientos técnicos avanzados.

Tabla 5.2: *Comparativa entre las diferentes herramientas para la captura de datos del CSI.*

5.2.2. Software

De los entornos creados con 4 GB de RAM, 2 núcleos, 40 GB de ROM y arquitectura AMD de 64 bits, testeamos las versiones: 24.04 LTS, 22.04 LTS, 20.04 LTS, 14.10 LTS, 14.04 LTS y 12.04 LTS. Descartamos todas las versiones superiores a la 14.10 debido a su incompatibilidad con el kernel modificado.

- **Máquinas virtuales con diferentes versiones de Ubuntu.** De los entornos creados con 4 GB de RAM, 2 núcleos, 40 GB de ROM y arquitectura AMD de 64 bits, testeamos las versiones: 24.04 LTS, 22.04 LTS, 20.04 LTS, 14.10 LTS, 14.04 LTS y 12.04 LTS. Descartamos todas las versiones superiores a la 14.10 debido a su incompatibilidad con el kernel modificado.
- **Herramientas para la captura del CSI.** Existen diversas herramientas para la captura de datos del CSI, como se muestra en la tabla 5.2, cada una con sus propias ventajas y desventajas.Como veremos a continuación, todas las opciones que hemos estudiado y valorado —a excepción de los SDR— están diseñadas para trabajar con hardware comercial, NICs comerciales que son capaces de capturar el CSI y, de hecho, lo suelen usar para su funcionamiento, pero que no permiten al usuario a acceder a dicha información. Así pues, haciendo unas modificaciones en el firmware se puede habilitar la extracción de CSI en dispositivos que no lo soportan de manera nativa.

  - **Linux 802.11n CSI Tool.** La descartamos porque el firmware y los controladores necesarios estaban diseñados para la tarjeta de red Intel 5300, que no es compatible con las placas base utilizadas.
    La consideramos inicialmente por presentar las siguientes características:
    - La Intel 5300 NIC reporta el CSI de 30 grupos de subportadoras, distribuidas uniformemente entre las 56 subportadoras de un Canal de 20

MHz o las 114 subportadoras de un canal de 40 MHz [20].
  - Permite capturar mediciones detalladas del Canal, que pueden emplearse en una amplia gama de aplicaciones, incluyendo la recopilación, lectura y análisis del CSI.
  - El código en espacio de usuario puede procesar los datos como parte de un algoritmo en línea o simplemente volcar la traza en disco para un procesamiento posterior.
  - Se proporcionan utilidades en MATLAB para analizar el formato estandarizado del CSI y calcular características estándar del Canal [19], lo que proporciona una comprensión básica de los canales inalámbricos MIMO y sirve como base para desarrollos futuros.

- **Nexmon CSI Tool.** La descartamos porque no era posible instalar un hardware compatible con Nexmon en la placa base de la tragaperras utilizada en nuestra investigación.
  Consideramos utilizarla dado que:
  - Es una herramienta de código abierto [24].
  - Es compatible con los dispositivos Raspberry Pi que incorporan el chipset BCM4339, con móviles como los Nexus 5 y 6P, y con routers de alta gama que soportan los chipsets BCM4365 y BCM4366 [25].
  - Soporta el canal de los 80 MHz, es decir, la frecuencia en la que trabaja el estándar 802.11ac (5G).

- **ESP32 CSI Tool.** La descartamos debido a las imprecisiones de los datos y a lo tedioso que resultó compilar el conjunto de herramientas.
  Los motivos por los que la consideramos fueron:
  - Permite la transmisión y recepción de información en tiempo real [27].
  - Ofrece la posibilidad de dejar la placa sola volcando los datos sobre una microSD para su procesamiento posterior [26].
  - Cuenta con una serie de herramientas prácticas que permiten la visualización en tiempo real del muestreo a través de dispositivos móviles [28] [29].

- **Software Defined Radio (SDR).** Un sistema de comunicación por radio definida por software (SDR) es un sistema en el que componentes tradicionales, como mezcladores, filtros, amplificadores y moduladores, se implementan mediante software en un ordenador o un sistema integrado. Decidimos no considerarlo para esta investigación debido a su complejidad y coste, con una inversión que puede oscilar entre 200€ y varios miles de euros. Además, su correcta utilización exige conocimientos avanzados en señales y sistemas de comunicación.
  Las características por las que lo consideramos fueron:
  - Gran flexibilidad y adaptabilidad ya que el mismo hardware se puede reconfigurar para manejar diferentes protocolos de radio simplemente

cambiando el software.

- Capacidad para recibir y transmitir una amplia gama de frecuencias y protocolos.

6
Metodología

El estudio se ha diseñado utilizando una metodología experimental, la cual hemos considerado estructurar en tres fases que desembocan en sucesivas pruebas de ensayo-error. La primera fase consiste en la búsqueda de material y la segunda en el diseño y ensamblaje del sistema, ambas se exponen a continuación. La última fase se desarrollará de manera más exhaustiva en el apartado de resultados.

6.1. Búsqueda de material y dificultades

Recordemos que, al seleccionar la herramienta Atheros CSI Tool, el sistema que vamos a diseñar estará formado por dos equipos que están en comunicación gracias a un tercero, el AP. Además, cada uno de los equipos necesita de una NIC. Los dos equipos, transmisor y receptor, están compuestos por una placa base de una tragaperras retirada de la marca GiGames y un portátil Toshiba que nos prestaron del instituto.
No obstante, antes de llegar a estos dos equipos se presentaron diferentes dificultades. En el caso de la placa base de la tragaperras GiGames, llegamos a utilizar hasta diez placas diferentes debido a que muchas de ellas estaban dañadas o en mal estado. En lo que respecta al portátil, realizamos diferentes pruebas con equipos disponibles en el hogar y solicitamos el acceso a otros ordenadores en el instituto antes de llegar al portátil de Toshiba.
El problema para encontrar este segundo equipo radicaba en que era necesario un dispositivo que fuera compatible con el estándar de bus Mini PCI Express (PCI-e), ya que así el mismo modelo de NIC nos servía tanto para el portátil como para la placa.
El siguiente paso consistía en determinar el modelo de NIC de Atheros que utilizaríamos. Los investigadores del estudio DensePose from WiFi evaluaron diversas tarjetas de esta marca e idearon una lista [13] en la que figuraban los modelos que habían probado y que eran compatibles, así como aquellos que, por similitudes también debían serlo. No obstante, la lista resultó ambigua, ya que existían numerosos modelos que aún no habían sido sometidos a pruebas.
En nuestras placas base se encontraban precisamente estos modelos por lo que fue-

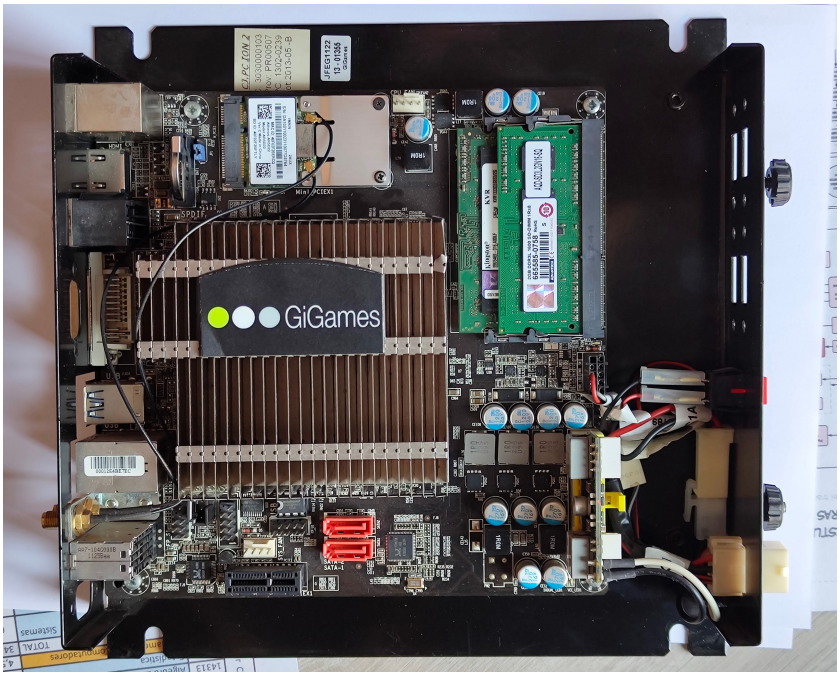


Figura 6.1: *GiGames motherboard.*

ron los que probamos inicialmente. Sin embargo, ambos modelos, identificados como AR9285 y AR9485, resultaron ser incompatibles con la herramienta de Atheros, lo que nos llevó a descartarlos como sujetos de estudio. Realizamos las mismas pruebas en un adaptador Wi-Fi USB que incorporaba el mismo chip que el modelo AR9462, el cual resultó ser compatible. Sin embargo, el resultado obtenido también fue desfavorable.
Al no conseguir resultados por la falta de una NIC compatible con la herramienta Atheros, nos decantamos por buscar el modelo AR9462, el cual sí estaba comprobada su compatibilidad.

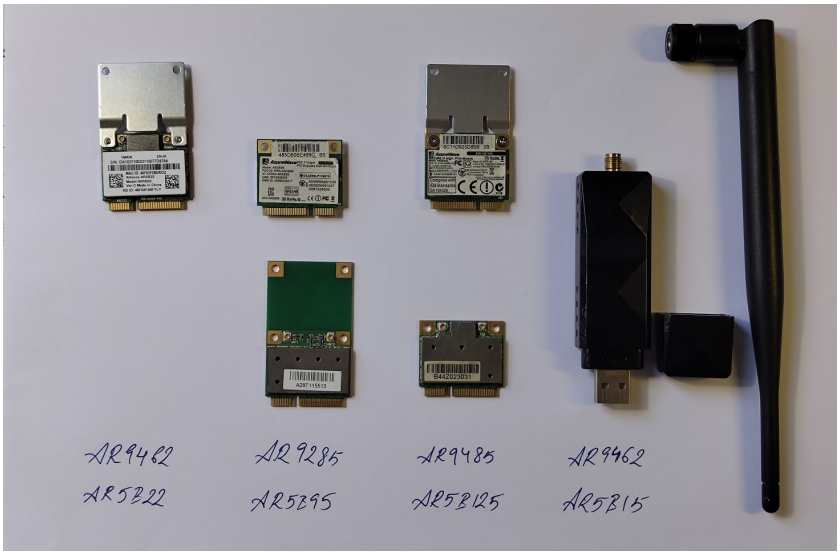


Figura 6.2: *Tarjetas Wi-Fi testeadas con el software Atheros CSI Tool.*

El inconveniente a la hora de conseguir este modelo concreto de Atheros fue que no

era posible adquirirlo individualmente, a pesar de estar presente en muchos equipos. Al acudir a diferentes tiendas de informática y electrónica, nos dimos cuenta de que no vendían tarjetas Wi-Fi de este tipo por separado, a menos que fuera para hacer una reparación o la compra de un equipo completo. Recordemos que este trabajo no dispone de fondos. En consecuencia, nos vimos obligados a abandonar la búsqueda local y terminamos adquiriendo dos de estas tarjetas a un vendedor alemán a través de eBay. Por último, respecto al punto de acceso, determinamos inicialmente la utilización de una Raspberry Pi —ya que disponíamos de varias unidades—. Finalmente, optamos por emplear mi router doméstico ya que, durante el desarrollo de la investigación, las placas de Raspberry se encontraban prácticamente agotadas y con un precio significativamente superior al habitual.
Es importante destacar que la configuración de hardware utilizada en este estudio puede modificarse a medida que se identifiquen mejoras o se aborden nuevos desafíos técnicos durante el desarrollo de la investigación.

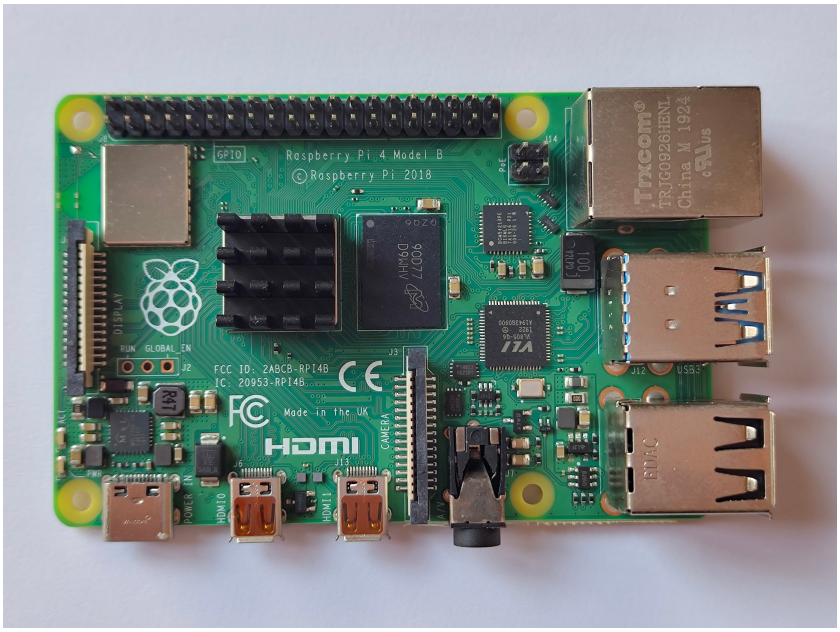


Figura 6.3: *Raspberry Pi 3B empleada en primera instancia como AP.*

6.2. Diseño y ensamblaje del sistema

6.2.1. Laboratorio de pruebas para el estudio del CSI con la herramienta Atheros

Una vez tuvimos claro el hardware que íbamos a utilizar, establecimos un entorno con el que poder trabajar y estudiar el CSI. En consecuencia, ubicamos los equipos de acuerdo con la configuración representada en la figura 6.4, estableciendo un arreglo triangular con el router ubicado en el vértice superior.
El motivo de emplear esta disposición, y no otra, era principalmente por tema de espa-

cio. Las tomas de corriente para el router y los dos equipos nos limitaban las opciones.

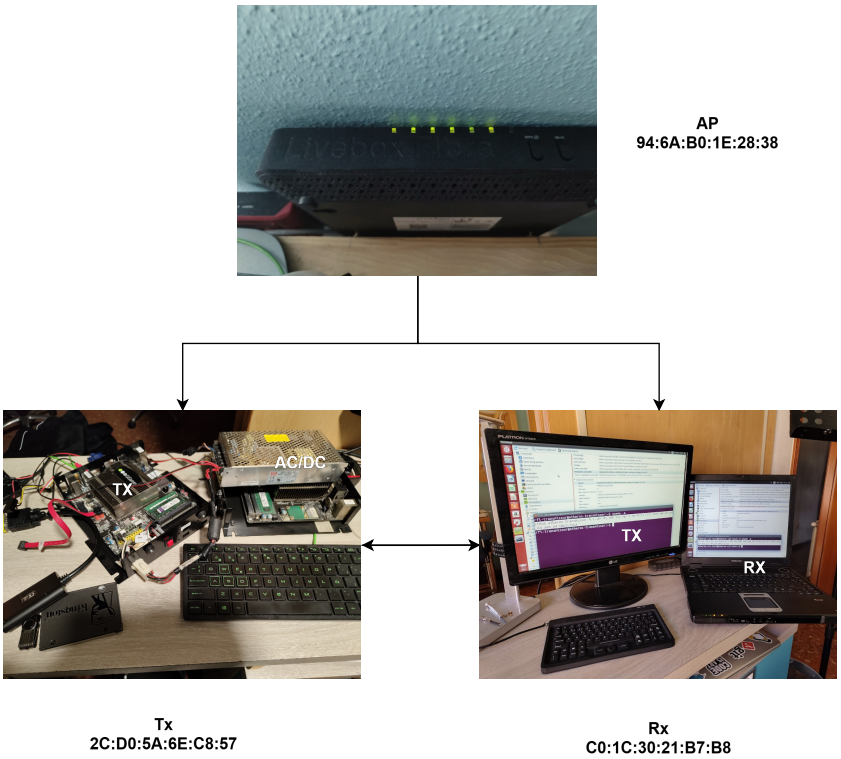


Figura 6.4: *Disposición de los diferentes equipos para el estudio del CSI con la herramienta Atheros.*

6.2.2. Adaptación y diseño de la réplica de DensePose From WiFi

La motivación que presidía la investigación era replicar el sistema desarrollado en el estudio DensePose From Wi-Fi con ciertas modificaciones.

En primer lugar, minimizamos los costes asociados al ensamblaje del entorno de pruebas mediante la utilización de hardware más accesible.

En segundo lugar, orientamos el estudio hacia un enfoque más general y centrado en el movimiento, al tiempo abordábamos las cuestiones de privacidad y ética planteadas.

Y, por último, tratamos de trasladar a la práctica todos aquellos conceptos matemáticos y físicos que excedían nuestro alcance.

Entrando en materia, la metodología que abordamos se articula en torno a los siguientes puntos:

- **Sanitización de la fase.** Dado que las señales Wi-Fi suelen contener ruido en su fase, implementamos un proceso de sanitización para estabilizar dichos valores y mejorar la precisión del modelo.
- **Traducción y correspondencia.** La red de traducción de modalidad emplea una red neuronal para traducir las señales Wi-Fi del dominio CSI a nuestro espacio 3D. Logramos esta traducción generando mapas de características similares a imágenes en 2D. Una vez que mapeamos los datos, hicimos uso de una arquitectura

similar a DensePose-RCNN[1] para predecir mapas UV y mapas de calor de puntos clave a partir de las señales Wi-Fi.

- **Eficiencia.** Empleamos una red preentrenada y basada en imágenes que sirvió de modelo y dirigió el entrenamiento de la otra red basada en Wi-Fi (3). Esta estrategia, ampliamente utilizada en el entrenamiento de modelos de redes neuronales que procesan grandes volúmenes de datos, resulta en una sustancial reducción de los tiempos de aprendizaje.

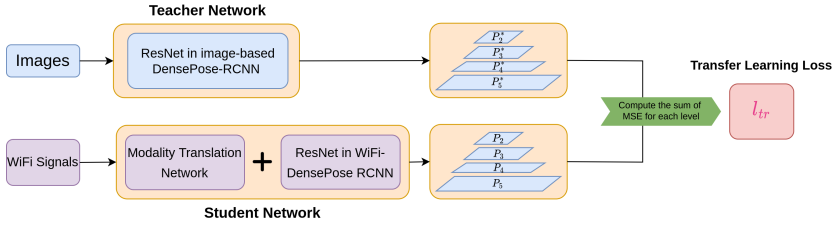


Figura 6.5: *Aprendizaje transferido de una red maestra basada en imágenes a la red basada en WiFi, WiFi-DensePose RCNN. Foto extraída del paper DensePose From Wi-Fi.*

Nos centramos entonces en la captura de las señales Wi-Fi en su estado original —en crudo—, es decir, tal y como se transmiten a través del aire.
Una vez obtenidas, aplicamos diversas transformaciones y tratamientos para extraer y representar gráficamente la amplitud y la fase de las señales capturadas empleando MATLAB.
Nos propusimos estudiar su comportamiento al colocar diferentes cuerpos en movimiento entre el receptor y el emisor, a la vez que comprobamos las hipótesis previamente planteadas. Una vez recopilados y realizados los estudios pertinentes, procedimos a la evaluación de la viabilidad de trasladar este sistema a un entorno en tiempo real. El objetivo no se limitó a la captura de imágenes, sino también al reconocimiento y procesamiento en tiempo real de datos, que posteriormente trataríamos y utilizaríamos para entrenar nuestra propia red neuronal convolucional basada en regiones.

6.3. Instalación, configuración y experimentación

Una vez que el sistema fue funcional, instalamos y configuramos el software necesario para capturar y calcular los datos del CSI. En esta fase intervendrán los equipos transmisor y receptor así como el programa Atheros CSI Tool. Posteriormente, interpretaremos los datos obtenidos con MATLAB realizando distintas pruebas a partir de los errores o dificultades presentadas tal y como veremos en la siguiente sección.

[1] DensePose-RCNN constituye una arquitectura de red neuronal diseñada para la estimación de la pose humana, fundamentada en la arquitectura de Mask-RCNN, incorporando las características de Feature Pyramid Network (FPN) y la agrupación ROI-Align, con el propósito de obtener marcadores de partes densas y coordenadas dentro de cada una de las regiones seleccionadas.

7 Resultados

7.1. Instalación del Sistema Operativo en los distintos dispositivos

Dado que en este estudio estamos empleando hardware asequible y rudimentario, necesitamos hacer uso de un sistema operativo (SO) liviano, capaz de correr en equipos como los descritos, sin consumir todos los recursos de estos e imposibilitando su uso. Además, este OS debe ser compatible con la herramienta de Atheros. De manera que las opciones que tenemos son tres versiones de Ubuntu: 12.04 LTS, 14.04 LTS y 14.10 LTS. De todas ellas, acabamos usando la versión 14.04 LTS debido a que fue la que mejor rendimiento nos otorgaba en las pruebas con las máquinas virtuales.

Como en cualquier instalación de otra distribución de Linux, necesitamos la imagen del SO. Al tratarse de una versión antigua, la descargamos del servidor old-releases.ubuntu.com. Una vez descargada, tenemos que bootear o grabar el USB con dicha imagen. Para ello, empleamos el comando `dd`:

Listing 1: *Comando para grabar una imagen ISO en una unidad USB utilizando* `dd`.

```
1 sudo dd if=archivo.iso of=/dev/sdX bs=4M iflag=fullblock status=progress
```

Los parámetros que recibe el comando son la ubicación de la imagen ISO (`if=…`), la ubicación del USB (`of=…`), el tamaño de cada bloque de datos (`bs=4M`), comprobar que cada lectura se complete con el tamaño total del bloque especificado (`iflag=fullblock`) y, por último, como el comando `dd` no muestra información por pantalla por defecto, el parámetro `status=progress` permite que se imprima el progreso en tiempo real.

Luego de tener el USB booteado y conectado al equipo, modificamos el orden de arranque por defecto en la BIOS de la CPU y el portátil para que arrancasen por el USB.

El proceso de instalación de Ubuntu es relativamente sencillo —siguiente, siguiente—, por lo que no lo abordaremos en este documento.

Figura 7.1: *Clásico panel de instalación de Ubuntu.*

7.1.1. Compilacón del kernel

El kernel es como el corazón de un sistema operativo. Este componente, ubicado entre la capa de aplicaciones y el hardware, es un elemento esencial al estar dotado de la capacidad de acceder de manera irrestricta a la totalidad de los componentes de un sistema. Facilita el acceso seguro a las distintas partes del hardware y se define como el proceso por el cual se establece una conexión entre la capa física, el hardware y la capa de software, también conocida como aplicaciones. Al ejecutar aplicaciones, se requiere del uso de recursos como espacio para almacenar información en disco o potencia gráfica o de cómputo por parte de la CPU. Es entonces cuando el kernel entra en juego, ya que es el encargado de gestionar el acceso a los recursos físicos necesarios para llevar a cabo dichas tareas en el sistema operativo. El kernel evalúa las solicitudes de acceso a los recursos y se asegura de que la comunicación entre el hardware y el software se realice de manera eficiente y segura.

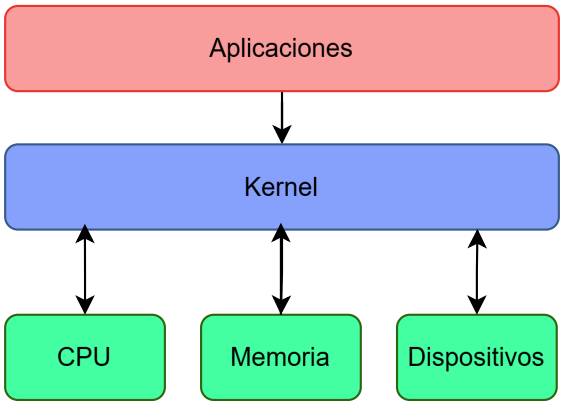


Figura 7.2: *Simplificación de como el kernel conecta a las aplicaciones (capa de software) con la capa física (hardware) de un ordenador.*

De manera predeterminada, la versión de Ubuntu que hemos instalado trae consigo la versión `4.4.02-142-generic` del kernel de Linux. Como se ha mencionado, el kernel se comunica con el hardware, para ello, necesitamos de la existencia de unos programas que le indiquen al sistema operativo cómo trabajar con la información que le devuelven dichos componentes físicos. Aquí es donde entran los módulos, también conocidos como drivers o controladores. Cada driver le dice al sistema operativo cómo debe comunicarse e interpretar los datos de determinado componente. Así pues, a nosotros nos interesa concretar al ordenador cómo debe comunicarse con la tarjeta de red de manera que esta nos brinde acceso a la información del CSI. A priori, esto no ocurre con el módulo que trae esta versión del kernel, así que instalaremos una versión modificada que cargue el módulo modificado para nuestra tarjeta de Atheros —el módulo en cuestión es el ath9k—. Esta versión del kernel modificada es la 4.1.10+. Para su instalación, sólo debemos bajarnos el código de su repositorio y compilarlo. Para más información de la instalación y la compilación, sus desarrolladores han publicado una guía paso a paso [16].

7.1.2. Error con la compilación del nuevo kernel

Puesto que estamos manejando un hardware modesto, la compilación del kernel la tuvimos que repetir en reiteradas ocasiones porque los equipos se quedaban congelados o había algún error durante la compilación que pensamos que estaba asociada a las limitaciones técnicas.

Una vez abordamos la instalación de este kernel, nos encontramos que, al reiniciar los equipos, la versión del kernel no había cambiado. Este problema se debía a que cuando tenemos varios kernels instalados en un sistema, el bootloader (cargador de arranque, software que se ejecuta antes del kernel y después de la BIOS) elegía como primera opción aquella versión del kernel que fuera más reciente, por lo que la versión modificada, a pesar de estar instalada, no se ejecutaba con el arranque del equipo. Para solucionar este inconveniente, modificamos el archivo encargado de la configuración de la GRUB (*GRand Unified Bootloader*) para que nos dejase elegir manualmente el kernel con el que queríamos arrancar el sistema.

Las líneas que modificamos dentro del `/boot/grub/grub.cfg` fueron las siguientes:

Listing 2: *Líneas modificadas del archivo de configuración de la grub para mostrar las opciones de arranque.*

```
1 GRUB_TIMEOUT_STYLE=menu  # al sólo haber un SO instalado, el menú de la GRUB permanece
  ↪ oculto
2 GRUB_TIMEOUT=10  # tiempo de espera para seleccionar una opción
```

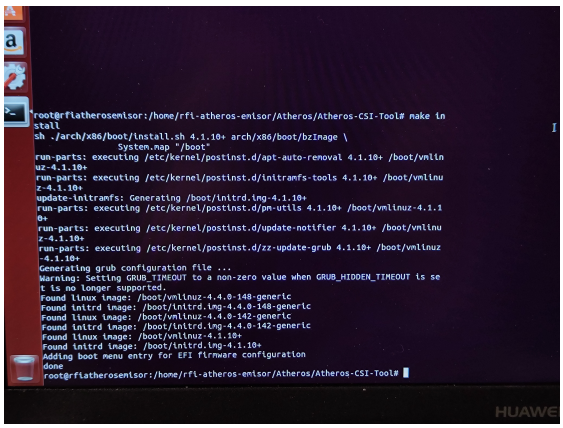


Figura 7.3: *Compilación exitosa del kernel 4.1.10+*

7.2. Atheros CSI Tool

En esta sección trataremos todo lo relacionado con el uso de las herramientas que nos ofrece el software Atheros CSI Tool para la recopilación y análisis del CSI, es decir, desde la comunicación entre el Tx y el Rx, pasando por la instalación de estos programas, la configuración previa para usar los scripts de transmisión y recepción de datos, y el análisis de los datos recopilados en MATLAB.

7.2.1. Comunicación entre el Tx y el Rx

En el momento de calcular el CSI —recordemos que el proceso varía según el medio por el que se transmiten las señales Wi-Fi—, existen diferentes formas de establecer la comunicación entre los equipos transmisor y receptor. En la figura 7.4 se muestran las disposiciones empleadas durante el desarrollo del estudio.
Aprovechando la disposición triangular de los tres equipos que hemos descrito en el punto sobre el ensamblaje nos encontramos con los siguientes modos:

- **Modo AP-Client**. En este esquema clásico, uno de los dispositivos actúa como punto de acceso y el otro como cliente. En el ámbito de la comunicación inalámbrica, se ha observado que únicamente un dispositivo puede transmitir de manera concurrente. En consecuencia, el receptor se ve encargado de determinar el CSI del paquete recibido. Este modo reproduce una red Wi-Fi convencional y resulta útil para ensayar la configuración básica y la sincronización de los equipos.
- **Modo Broadcast.** Este modelo es similar al anterior en cuanto a estructura, pero con una diferencia clave: todos los clientes comparten la misma dirección MAC. Este fenómeno permite que todos los receptores escuchen los paquetes emitidos por el AP de manera simultánea, lo que abre la puerta a la captura del CSI por múltiples receptores al mismo tiempo. Esto resulta especialmente útil en espacios donde se analiza la influencia de la señal desde distintos puntos.
- **Modo Monitor-Injection.** En contraste con los métodos anteriores, en este modo no se establece una red convencional con puntos de acceso y clientes. En su

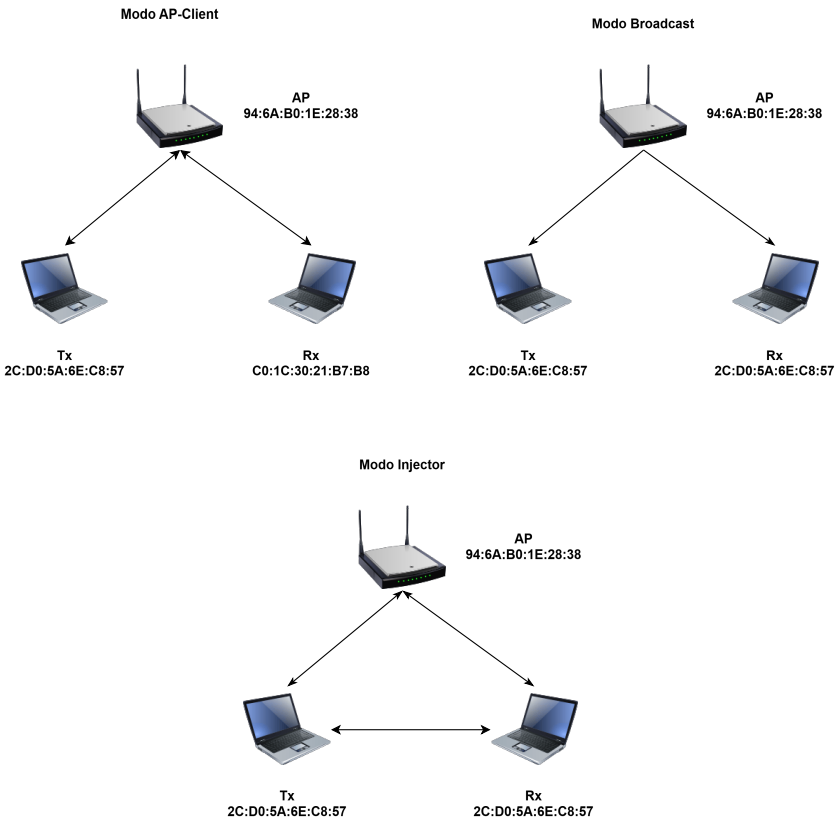


Figura 7.4: *Disposiciones y esquema de roles de cada equipo durante la captura del CSI.*

lugar, uno o varios dispositivos «inyectan» paquetes directamente en el medio inalámbrico —"lanzan" la información al aire—, mientras que otros los capturan en modo monitor —los "recogen" del aire—. Este enfoque elimina la necesidad de asociación entre dispositivos y nos permite un mayor control sobre las Tramas transmitidas. Si bien esta funcionalidad se encuentra más desarrollada en sistemas que operan con el router de OpenWRT, logramos una configuración básica en Ubuntu mediante la creación de una interfaz en modo monitor.

Aunque hicimos uso de los tres enfoques para evaluar la diferencia entre ellos, en la práctica empleamos el modelo clásico de cliente-AP por convención y facilidad de uso ya que obteníamos los mismos resultados.

7.2.2. Instalación Atheros CSI Tool

Llegados a este punto donde tenemos claro cómo recolectar el CSI en nuestro laboratorio de trabajo, necesitamos instalar la mencionada herramienta Atheros CSI Tool. Esta aplicación de código abierto que se encuentra alojada en el repositorio de GitHub *Atheros-CSI-Tool-UserSpace-APP*, incluye una serie de ejecutables o scripts cuya funcionalidad es establecer la comunicación entre ambos equipos.
Por una parte, dentro del directorio `recvCSI/` encontramos el módulo escrito en C que, al compilarse, nos crea el binario `./recv_csi`. Este ejecutable únicamente recibe un parámetro: la ubicación del archivo donde se volcarán los datos de la captura.

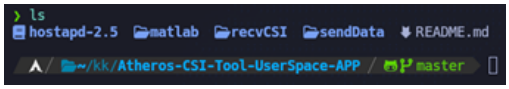


Figura 7.5: *Listado de archivos del repo Atheros-CSI-Tool-UserSpace-APP.*

Listing 3: *Parámetros que recibe el binario* `recv_csi`.

```
./recv_csi output_file.dat
```

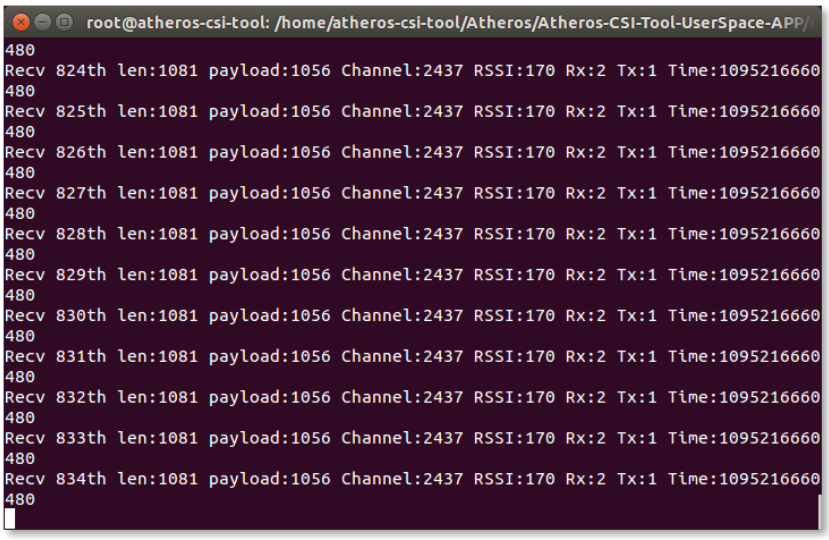


Figura 7.6: *Salida del comando* `recv_csi` *al recibir los paquetes HT del emisor.*

En cuanto al equipo transmisor, dentro del directorio `sendData/` encontramos su módulo correspondiente, donde al compilar los archivos, tenemos acceso al script `./send_Data`, que recibe 4 argumentos: la interfaz de red que enviará los datos que posteriormente servirán para calcular el CSI, la dirección MAC del receptor y el número de paquetes que se enviarán. Cuantos más paquetes se envíen, más tiempo transcurrirá y el archivo de la captura será más grande.

Listing 4: *Parámetros que recibe el binario* `send_Data`.

```
./send_Data ifName DstMacAddr NumOfPacketToSend
```

7.2.3. Preparación de Scripts de transmisión y recepción

Previamente a la comunicación entre equipos, debemos activar un mensaje que será emitido por el emisor y recibido por el receptor. Es preciso señalar que la presente comunicación se centra en el mensaje previamente acordado. Este mensaje recibe la denominación de «flag», y su valor debe ser establecido como «true», 1 en sistema binario. Esta información la obtuvimos a través de una conversación mantenida en el repositorio *Atheros_csi_Tool_OpenWRT_src* en GitHub con el usuario geosearchef, inge-

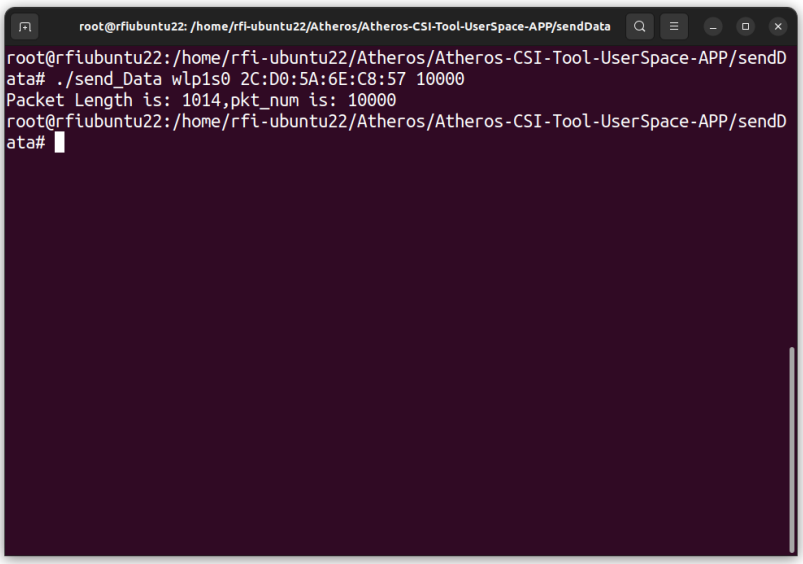


Figura 7.7: *Salida del comando* `send_Data`.

niero informático de la universidad de Munich. La conversación, de carácter público, se halla disponible en la issue n. 62 [30]. Como conclusión a la conversación mantenida, procedimos con la revisión de la configuración del driver `ath9_k`, ubicado en `/drivers/net/Wireless/ath/ath9k/ath9k.h`, con el propósito de verificar que la flag estuviera predefinida y activada.

En efecto, así fue. Observamos que la flag estaba seteada en 1. Este fenómeno se atribuye a la configuración predeterminada del kernel modificado, en el cual el valor se establece verdadero por defecto.

En segundo lugar, y como parte del proceso de preparación del sistema, fuimos ajustando el valor del buffer en múltiplos de 2^n bytes en el archivo `sendData/sendData.c`. El buffer constituye un espacio en memoria destinado al almacenamiento temporal de información puntual durante diversos procesos de intercambio de datos, con el propósito de prevenir la pérdida de información. La finalidad de esta modificación era garantizar la capacidad del sistema para gestionar el envío de grandes volúmenes de paquetes, mitigar el riesgo de un posible desbordamiento del buffer y asegurar la integridad y la estabilidad del sistema en su conjunto.

Listing 5: *Constante modificada del script* `sendData.c`.

```
1 #define BUF_SIZ     2048
```

De forma paralela, para el archivo `recvCSI/csi_fun.c` modificamos la constante `csi_st_len` definida como `#define csi_st_len 23` para localizar campos de información específica en un buffer (`buff_addr`). Antes estaba en null. Este valor se utiliza como punto de referencia para analizar la estructura de datos CSI dentro del buffer recibido.

Por defecto, el buffer receptor/recibido (`buff_addr`), usado para comunicarse con el

kernel, viene establecido en la siguiente ubicación: `unsigned char buf_addr[4096];`. Si el kernel recibe un paquete y calcula su CSI, los valores del CSI, así como el payload y la información del estado del proceso, se volcarán sobre este buffer. Por ello, es importante que esta ubicación esté libre y reservada únicamente para el script de la suite de Atheros para el receptor.
Una vez reseteadas estas variables, la función encargada de escuchar y leer el CSI del kernel: `int read\_csi_buf(unsigned char* buf_addr,int fd, int BUFSIZE)` leerá inmediatamente el buffer después de que el kernel reciba un paquete y calcule el CSI. Además, existe un temporizador que, al llegar a cero, obligará a la función a leer el buffer haya o no recibido el CSI del kernel. La salida de la función devolverá un entero que indica el número de bytes que conforman el buffer. Si el número es 0, significa que el CSI no se ha subido.

7.2.4. Paquete de sondeo HT. Mensaje pactado

En la anterior subsección hemos hablado del mensaje previamente acordado. Ahora, entraremos en detalle en la estructura y la conformación de este paquete.
El nombre técnico que recibe es paquete de sondeo o *HT* (*High Throughput*) *Sounding Packet*. En redes Wi-Fi modernas —especialmente IEEE 802.11n/ac/ax— los paquetes de sondeo son Tramas especiales que permiten a un transmisor medir la calidad del canal inalámbrico entre él y uno o más receptores. El objetivo principal de enviar un paquete de este tipo es que el receptor pueda observar cómo las ondas Wi-Fi se ven afectadas por el entorno —reflexiones, difracciones, dispersiones, interferencias, etc.—, y devolver información precisa sobre el estado del canal.
En el caso de redes HT (802.11n), estos paquetes permiten al transmisor optimizar la comunicación utilizando técnicas avanzadas como MIMO, donde interactúan múltiples antenas; *beamforming*, enfocado en focalizar la señal de manera inteligente hacia un receptor [33]; o calcular matrices semicomplejas que describen en detalle cómo se modifica cada subportadora OFDM (*Orthogonal Frequency-Division Multiplexing*) durante la transmisión de información. Justo lo que a nosotros nos interesa. Todo ello, es gracias a que cada uno de estos paquetes transmitidos contienen una flag conocida por el Tx y el Rx, de manera que el receptor es capaz de estudiar las diferencias entre la señal Señal piloto (Tx) —señal transmitida por el emisor— y la señal testigo —recibida por el receptor—. Como hemos visto a lo largo de esta investigación, los datos proporcionados por esta variación pueden otorgar información realmente útil que luego puede ser enviada de vuelta al emisor y, a partir de ahí, abordar el envío de información de una manera más eficiente al conocer el estado del canal.
Con el objetivo de estudiar el CSI, a partir de este paquete y de su posterior cálculo, nos encontramos con que los paquetes de las subportadoras que transportan el CSI usan arrays de números complejos para almacenar las matrices del CSI empleando un máximo de 3 antenas emisoras y 3 receptoras con 114 subportadoras para el Canal de los 40MHz. Como nosotros usamos tan solo 4 antenas —2 Tx y 2 Rx–, la matriz no se

ocupó por completo.
La estructura del paquete es entonces la siguiente:

Listing 6: *Información enviada por el emisor en cada Trama.*

```
typedef struct
{
u_int64_t
tstamp;
/* h/w assigned time stamp */
u_int16_t
u_int8_t
u_int8_t
u_int8_t
u_int8_t
u_int8_t
channel;
chanBW;
rate;
nr;
nc;
num_tones;
/* wireless channel (represented in Hz)*/
/* channel bandwidth (0->20MHz,1->40MHz)*/
/* transmission rate*/
/* number of receiving antenna*/
/* number of transmitting antenna*/
/* number of tones (subcarriers) */
u_int8_t
u_int8_t
noise;
phyerr;
/* noise floor (to be updated)*/
/* phy error code (set to 0 if correct)*/
u_int8_t
u_int8_t
u_int8_t
u_int8_t
rssi;
rssi_0;
rssi_1;
rssi_2;
/*
 rx frame RSSI */
/*
 rx frame RSSI [ctl, chain 0] */
/*
 rx frame RSSI [ctl, chain 1] */
/*
 rx frame RSSI [ctl, chain 2] */
u_int16_t
 payload_len;
```

```
48  /*
49  payload length (bytes) */
50 u_int16_t
51  csi_len;
52  /*
53  csi data length (bytes) */
54 u_int16_t
55  buf_len;
56  /*
57  data length in buffer */
58 } csi_struct;
```

Field	Description	Units
timestamp	Marca de tiempo (TSF) de cuando se recibió el paquete	μs
csi_len	Longitud de de la información del CSI	Bytes
channel	Canal en el que se capturó la señal	MHz
err_info	Devuelve el código de estado de la ejecución	
noise_floor	Expresa el nivel del ruido (pero está seteado a 0 por defecto)	dB
Rate	Velocidad de datos del paquete recibido. Su valor es un valor absoluto entero	Bits
bandWidth	Ancho de banda del Canal. 20MHz si se establece en 0 y 40MHz si se establece en 1	MHz
num_tones	Número de subportadoras utilizadas para la transmisión de datos	
nr	Número de antenas receptoras	
nc	Número de antenas transmisoras	
rssi	Potencia de la señal de la combinación de todas las antenas activas	dBm
rssi1, rssi2, rssi3	Potencia correspondiente a cada antena activa	dBm
payload_len	Longitud de la carga útil del paquete recibido	Bytes
csi	Los datos del CSI en sí	
payload	Datos útiles transmitidos	

Tabla 7.1: *Información del estado capturada y almacenada a partir de cada Trama.*

De todos estos campos de información, a la hora de estudiar los cambios producidos en un entorno real, es decir, el movimiento en sí, podemos destacar dos opciones: el RSSI y el CSI.

La información que proporciona el RSSI (*Received Signal Strength Indicator*) es la fuerza de la señal recibida, producto de la combinación de todas las antenas activas. Con ella, podemos estimar la calidad de la comunicación entre un dispositivo y un punto de acceso. A medida que el valor de la intensidad se acerque a cero, mayor será la cobertura o, lo que es lo mismo, se experimentará una mejora en la recepción de la señal. Ahora bien, lo que determina la oscilación de los valores de la intensidad es el entorno —la distancia del emisor al receptor, la atenuación, los obstáculos—, de manera que nos está describiendo la naturaleza del Canal de comunicación pero de forma general. Esto es un problema porque no nos sirve para distinguir por qué la señal es baja —¿es por interferencias?, ¿por distancia?, ¿por *mutli-path*?—. Tampoco nos permite hacer estimaciones precisas de fases, trayectorias múltiples o modulaciones complejas a la vez de que no podemos reconstruir el Canal por completo.

En cuanto al CSI, este nos describe cómo una señal se propaga de un emisor a un receptor con precisión, proporciona datos más estables en entornos complejos y, a su vez, permite extraer información de cada subportadora, permitiendo reconstruir el Canal

por el que ha viajado en prácticamente su totalidad.
Después de haber volcado los datos de la señal en crudo sobre un archivo con la herramienta de Atheros, procedimos con su análisis en MATLAB, obteniendo celdas con la información del estado del Canal como se puede apreciar en al figura 7.8.

csi{1,1} × csi ×

1x1 struct with 17 fields

Field	Value	Size	Class
timestamp	3.6409e+09	1x1	double
csi_len	0	1x1	double
channel	2462	1x1	double
err_info	0	1x1	int32
noise_floor	0	1x1	int32
Rate	143	1x1	int32
bandWidth	1	1x1	int32
num_tones	114	1x1	int32
nr	2	1x1	int32
nc	1	1x1	int32
rssi	170	1x1	int32
rssi1	57	1x1	int32
rssi2	40	1x1	int32
rssi3	128	1x1	int32
payload_len	1040	1x1	double
csi	0	1x1	double
payload	*1040x1 uint8*	1040x1	uint8

Figura 7.8: *Campos de información para la celda {1,1}.*

Sin embargo, como podemos observar, los valores que obtenemos para el CSI son 0. Como hemos visto anteriormente, esto se corresponde con la salida de la función encargada de leer los datos del CSI del kernel. El 0 significa que el temporizador ha llegado a cero y la función ha volcado los datos proporcionados por el kernel en ese período de tiempo, es decir, ninguno.
Lamentablemente, es en este punto preciso donde nuestra investigación ha experimentado un cese en su avance. Nuestros conocimientos en el ámbito de la informática y el procesamiento de señales son limitados, lo que ha imposibilitado la identificación de una causa determinante que explique por qué los valores del CSI que registramos son invariablemente nulos. No obstante, se ha llevado a cabo una evaluación de diversas causas probables.
Así pues, pensamos que el error puede encontrarse en que la flag está mal configurada, de manera que por eso los valores del CSI devuelven 0. Para comprobar esta hipótesis, hicimos uso de un tercer equipo ejecutando el programa Wireshark —analizador de tráfico de red—, que colocamos deliberadamente entre medias de la comunicación del transmisor y el receptor. De esta forma, fuimos capaces de capturar los paquetes que viajaban por el aire de un equipo a otro.
Los resultados que obtuvimos fueron los siguientes:
Podemos ver que el contenido de los paquetes son secuencias de “aa”, que viene a ser como si fuese todo 1s o 0s, es decir, no aporta información útil alguna. Además, se menciona un posible error: “Bogus IPv4 version (10, must be 4)” que aparece en todas

Figura 7.9: *Monitor mostrando el tráfico de red capturado por Wireshark junto a un Baby Yoda con una antena Wi-Fi omnidireccional SMA-M.*

las Tramas. Consultando diferentes fuentes y preguntando en varios foros, llegamos a que otra persona llegó al mismo error que nosotros [31]. Sin embargo, tampoco arribó a ninguna solución o justificación y dio por cerrado el tema.

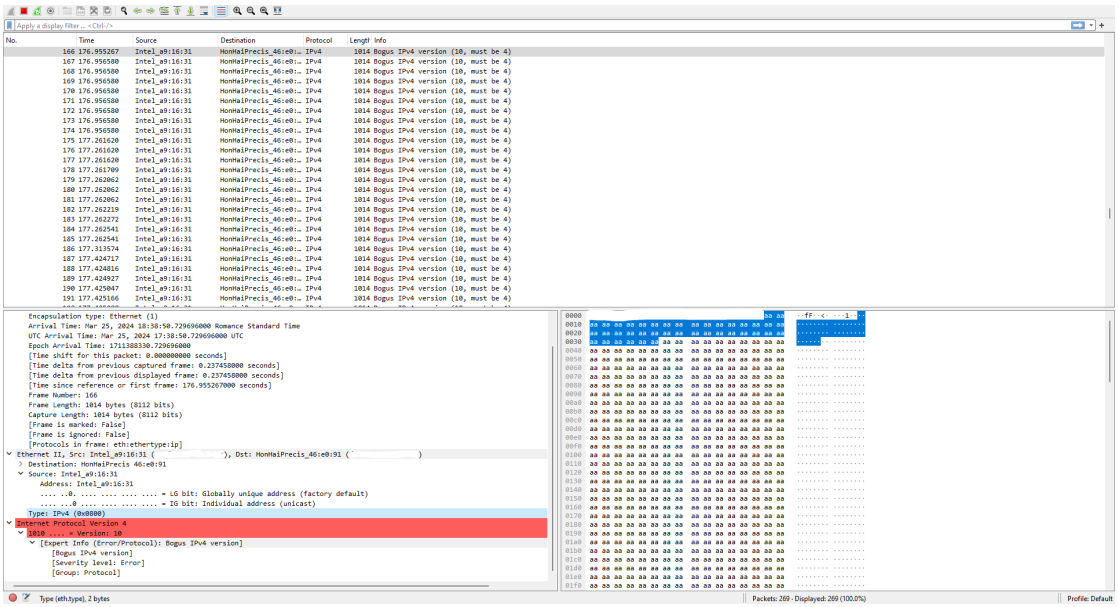


Figura 7.10: *Captura de Wireshark de los paquetes de sondeo interceptados enviados desde el emisor hasta el receptor.*

7.3. MATLAB

Una vez capturados los datos, estos pueden almacenarse y analizarse posteriormente con software especializado como MATLAB o Python, lo que permite procesar la información, extraer patrones y estudiar variaciones en el Canal inalámbrico.
Además, la capacidad de procesar los datos en tiempo real con Python o de realizar un análisis posterior con MATLAB nos ofrece la posibilidad de realizar una investigación más exhaustiva, al permitirnos el acceso a los datos en su estado original.
En esta sección, expondremos los scripts de MATLAB proporcionados en el repositorio *Atheros CSI Tool* para el análisis del CSI. Además, también desarrollamos nuestros propios scripts para trabajar con los datos que nosotros pudimos obtener.

7.3.1. Scripts para MATLAB de la suite de *Atheros CSI Tool*

Una vez capturadas las señales Wi-Fi —en *raw*— y habiéndolas almacenado en un archivo, lo subimos a MATLAB para su análisis.
En primera instancia, dentro del directorio `matlab/` del repositorio *Atheros CSI Tool* tenemos el programa `read_log_file.m` que es el encargado de extraer los campos de información mostrados en la tabla 7.1. Lo que podemos ver al ejecutar el script, es que se nos genera un espacio de trabajo con un array llamado `ans` de $n \times 1$ celdas, donde n es el número de columnas y dependerá de cuán grande sea la captura. Sin embargo, como ya han podido ver en la figura 7.8, el valor del campo `csi` para el elemento $\{1, 1\}$ es 0. Para no comprobar elemento por elemento en busca de un valor distinto a cero, desarrollamos el siguiente código que emplea un plot para graficar los valores del CSI en cada celda. El código se encuentra en 7 y la representación gráfica en 7.11. Pero, nuevamente, los valores siguen siendo 0.

Listing 7: *Código del script `csi_extract.m`.*

```
csi = read_log_file('records/output00.csi');

csiData = [];
time = [];

for i = 1:numel(csi)
    % Extraer la matriz de CSI de cada celda
    csiMatrix = csi{i}.csi;

    % Verificar las dimensiones de la matriz de CSI
    if isempty(csiData) || size(csiMatrix, 2) == size(csiData, 2)
        % Concatenar la matriz de CSI si es la primera iteración o si las dimensiones
        ↪ son consistentes
        csiData = [csiData; csiMatrix];
        time = [time; csi{i}.timestamp];
    else
        % Si las dimensiones no son consistentes, mostrar un mensaje de advertencia
```

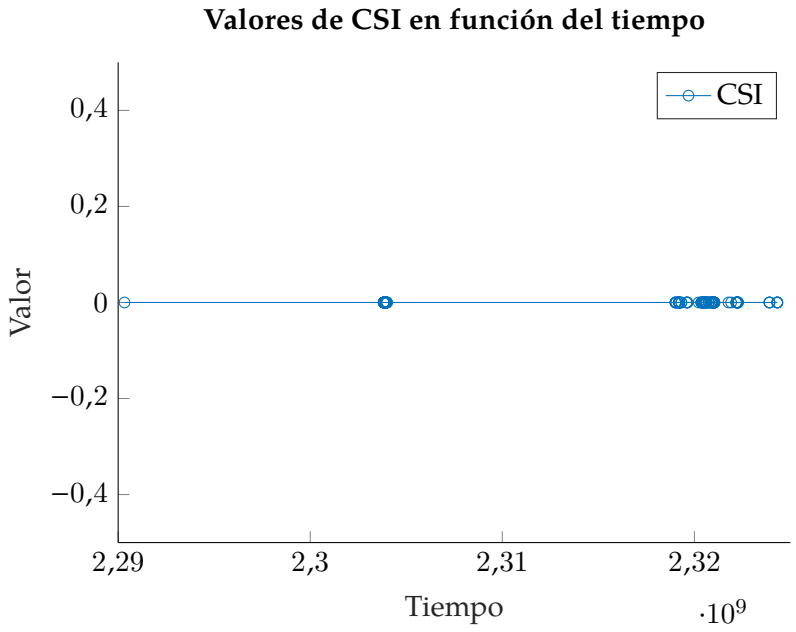


Figura 7.11: *Valores de CSI en función del tiempo.*

```
        warning(['Las dimensiones de la matriz de CSI en la iteración ', num2str(i), '
        ↪  no son consistentes. La matriz no se concatena.']);
    end
end

% Graficar los datos de CSI en función del tiempo
figure;
hold on;
plot(time, csiData, 'o-', 'DisplayName', 'CSI');
xlabel('Tiempo');
ylabel('Valor');
title('Valores de CSI en función del tiempo');
legend('Location', 'best');
hold off;

```

7.3.2. Representación gráfica de los datos obtenidos

Puesto que no tenemos información del CSI, el archivo `read_log_file.m` no nos proporciona mucha más información además de la ya observada. Así pues, decidimos escribir un programa que a partir los diferentes datos de la potencia de cada antena, los grafique en función del tiempo.

Listing 8: *Código del archivo `timeRSSI.m`.*

```
csi = read_log_file("records/output00.csi");

% Inicializar vectores para almacenar los valores de RSSI y tiempo
```

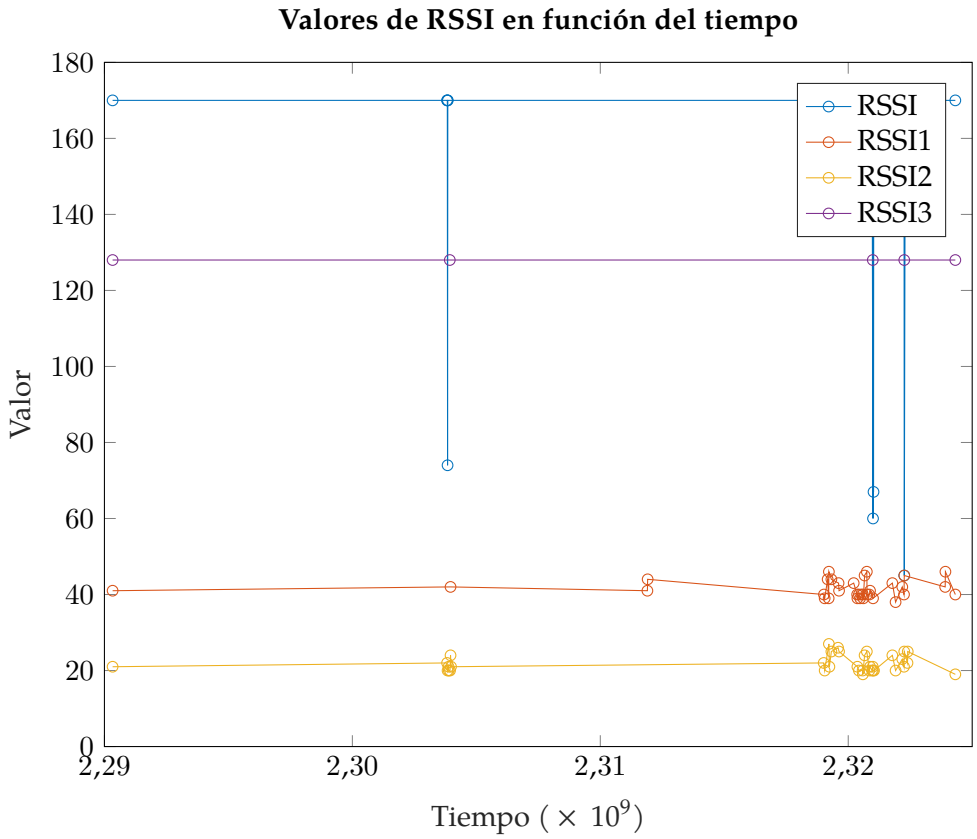


Figura 7.12: *Valores de RSSI de cada antena en función del tiempo.*

```
rssi_y = [];
rssi1 = [];
rssi2 = [];
rssi3 = [];
time_x = [];

% Iterar sobre todas las celdas del csi
for i = 1:numel(csi)
    % Extraer los valores de rssi y tiempo de cada celda y agregarlos a los vectores correspondientes
    rssi_y = [rssi_y; csi{i}.rssi];
    rssi1 = [rssi1; csi{i}.rssi1];
    rssi2 = [rssi2; csi{i}.rssi2];
    rssi3 = [rssi3; csi{i}.rssi3];
    time_x = [time_x; csi{i}.timestamp];
end

% Crear el gráfico lineal con cuatro conjuntos de datos
figure;
hold on;

% Graficar los datos para RSSI
plot(time_x, rssi_y, 'o-', 'DisplayName', 'RSSI');

```

```
% Graficar los datos para RSSI1
plot(time_x, rssi1, 'o-', 'DisplayName', 'RSSI1');

% Graficar los datos para RSSI2
plot(time_x, rssi2, 'o-', 'DisplayName', 'RSSI2');

% Graficar los datos para RSSI3
plot(time_x, rssi3, 'o-', 'DisplayName', 'RSSI3');

xlabel('Tiempo');
ylabel('Valor');
title('Valores de RSSI en función del tiempo');
legend('Location', 'best');
hold off;
```

8

Discusión y crítica de los aspectos éticos y morales

Cada día que nos despertamos y echamos una mirada atrás en el tiempo, podemos apreciar como nuestro modelo de vida, nuestra forma de pensar y las sociedades que conforman el mundo, han ido evolucionando y adaptándose a los tiempos que corren. Por lo general, todos estos cambios siguen una trayectoria lineal en el tiempo. Sin embargo, si acotamos la línea temporal a este último siglo, podemos apreciar una evolución con tendencia exponencial. Fenómeno que se explica por el vertiginoso avance tecnológico. Tan acelerado que resulta irrefrenable.

A lo largo de la historia, cada avance tecnológico ha traído consigo no sólo nuevas posibilidades, sino también nuevos retos éticos y sociales. La tecnología de detección mediante señales Wi-Fi no es una excepción. Si bien su desarrollo puede dirigirse hacia aplicaciones enfocadas en el bienestar y la conveniencia —como son la detección de caídas en personas de avanzada edad, la detección y reconocimiento de movimientos para implementaciones domóticas o, incluso, el rescate de personas en situaciones de emergencia—, también plantea interrogantes sobre la privacidad, las libertades individuales y el control en las sociedades contemporáneas.

Hoy en día, vivimos rodeados de sistemas de vigilancia cada vez más sofisticados: cámaras de seguridad en calles y edificios [34], porteros inteligentes [35], sistemas LiDAR en vehículos y drones [36], y tecnologías de reconocimiento facial instaladas en aeropuertos y espacios públicos como centros comerciales [37].

Todas estas herramientas, instaladas bajo la premisa de proteger al ciudadano o para hacerle frente al terrorismo, escondían deliberadamente el objetivo de escrutar las conductas de una población ajena. Sin embargo, hoy en día esta práctica es bien sabida y, aún así, hemos decidido normalizar la vigilancia masiva en gran medida. No obstante, está demostrado que cuando somos conscientes de que estamos siendo observados, tendemos a autocensurarnos por miedo a ser juzgados. Las cámaras son objetos físicos, tangibles y visibles —algunas incluso tienen la clásica lucecita roja— pero, ¿qué sucede

con las señales Wi-Fi?, ¿alguien ha sido capaz de ver alguna con sus ojos? Pese a ello, están en dondequiera que vaya. De hecho, es muy probable que mientras lea estas líneas esté siendo atravesado por multitud de ondas Wi-Fi en todas las direcciones —a menos que se encuentre en medio del bosque o durante un colapso general del sistema eléctrico—.

Por contraposición, los sistemas de detección basados en Wi-Fi suponen un cambio todavía más silencioso e imperceptible. Gracias a su capacidad para penetrar paredes y operar sin necesidad de contacto visual, permiten detectar la presencia y el movimiento de personas incluso en espacios privados, sin que estas sean conscientes de ello. Esta particularidad plantea una confrontación y una disyuntiva ética respecto a las disciplinas de vigilancia actuales. Nosotros somos los responsables de alimentar y dar forma a nuestra huella digital en cuanto a la información que publicamos en internet sobre nosotros. Sin embargo, no tenemos potestad en rehuir a ser grabados caminando por la calle, comprando en un centro comercial o siendo captado por la mirilla inteligente del vecino [38] mientras accede al apartamento de "Paco Fontanero" y, después de una tendida sobremesa, sale con los pelos desgreñados.

Ante estas coyunturas, la tecnología Wi-Fi emerge como una alternativa menos invasiva puesto que no es posible correlacionar los datos de las señales Wi-Fi con los rasgos característicos que permiten identificar a una persona.

Además, las ventajas técnicas son claras. La infraestructura Wi-Fi ya está desplegada en la mayoría de entornos, sólo se requiere del diseño del software encargado de permitir el estudio del CSI en cada uno de estos dispositivos que puede levantar redes Wi-Fi.

Pese a ello, existen riesgos que debemos considerar. El hecho de utilizar estas técnicas para vigilar sin consentimiento en oficinas, comercios o viviendas plantea serias amenazas a la privacidad individual. A diferencia de las cámaras, que pueden ser detectadas a simple vista, las ondas Wi-Fi son imperceptibles. Ello puede acarrear una sensación o sentimiento de *pseudoprivacidad*, que facilitaría la normalización de su uso sin un debate público informado.

Nos encontramos así ante una paradoja, mientras que la tecnología Wi-Fi de detección promete beneficios importantes, también puede ser utilizada como un arma de control social encubierta. La inexistencia de regulaciones claras sobre su uso abre la puerta a posibles abusos por parte de empresas y gobiernos, así como de cualquier curioso que tenga ciertos conocimientos en tratamiento y análisis de señales.

Es por ello que, con este trabajo de investigación que explora las capacidades técnicas de una tecnología relativamente nueva, no nos vemos capaces de abandonar los aspectos éticos y morales que puede llegar a presentar, a la vez que abogamos la necesidad perentoria de establecer marcos éticos y legales que protejan a los ciudadanos. Es crucial garantizar que estas herramientas se utilicen con total transparencia y respeto hacia los derechos fundamentales, evitando que su normalización siga el mismo camino que tecnologías ya cuestionadas como el reconocimiento facial con los sistemas basados en luz.

La tecnología Wi-Fi de detección, como toda herramienta creada por el ser humano, puede mejorar nuestro bienestar o erosionar nuestras libertades. Todo dependerá de nosotros como sociedad y nuestra consciencia sobre privacidad.

9
Conclusión

Pese a los resultados finales obtenidos, hemos podido explorar e investigar cómo una tecnología presente en prácticamente cualquier lugar, como es el Wi-Fi, podría usarse para estudiar el entorno en busca de patrones relacionados con el movimiento y la presencia humana, abriendo la puerta a innumerables aplicaciones de domótica y de bienestar. Todo ello empleando hardware accesible y presente en nuestros dispositivos domésticos junto con software libre, mientras estudiábamos sus implicaciones en nuestra vida y su repercusión en la sociedad hasta su posible perfeccionamiento e instauración.

Hemos demostrado el gran potencial del CSI, pero también hemos resaltado las limitaciones técnicas que aún deben resolverse. A futuro, esperamos poder ver el desarrollo de aplicaciones de software capaces de ejecutarse sin problema en diferentes equipos de bajos recursos y sin la necesidad de usar hardware especializado. Aunque este proyecto no logró todos los objetivos previstos, los aprendizajes obtenidos y las posibles líneas de investigación abren un camino prometedor para el uso de las redes inalámbricas en aplicaciones de eventos en tiempo real. El hecho de que este estudio pueda llegar a servir para continuar nuestra línea de trabajo, será un objetivo más cumplido.

Bibliografía

[1] Cadence System Analysis. (s.f.). Fundamentals of MIMO Communication in Wireless Systems [Consultado el 12 de marzo de 2024]. https://resources.system-analysis.cadence.com/blog/fundamentals-of-mimo-communication-in-wireless-systems

[2] Pérez, E. (s.f.). El reconocimiento facial de Mercadona acaba en multa de 2,5 millones de euros: qué dice la Agencia de Protección de Datos y qué lecciones se pueden extraer [Consultado el 26 de marzo de 2024]. https://www.xataka.com/privacidad/reconocimiento-facial-mercadona-acaba-multa-2-5-millones-euros-que-dice-agencia-proteccion-datos-que-lecciones-se-pueden-extraer

[3] Naciones Unidas. (2022, septiembre). Instrumentos de espionaje y vigilancia: Aumentan las amenazas a la intimidad y los derechos humanos, advierte un informe de las Naciones Unidas [Consultado el 16 de diciembre de 2023]. https://www.ohchr.org/es/press-releases/2022/09/spyware-and-surveillance-threats-privacy-and-human-rights-growing-un-report

[4] Samczynski, P., & Giusti, E. (Eds.). (2021, julio). *Recent Advancements in Radar Imaging and Sensing Technology* (Vol. 394). MDPI. https://doi.org/10.3390/books978-3-0365-0919-8

[5] Guarnera, C. (2024, enero). Aerial LiDAR 101: Introduction to its Applications and Benefits [Consultado el 8 de enero de 2024]. https://www.bluefalconaerial.com/aerial-lidar-101-an-introduction-to-its-applications-and-benefits/

[6] Ohno, M., Ukyo, R., Amano, T., Rizk, H., & Yamaguchi, H. (2023). Privacy-preserving Pedestrian Tracking using Distributed 3D LiDARs. *arXiv preprint arXiv:2303.09915*.

[7] Geng, J., Huang, D., & la Torre, F. D. (2022). DensePose From WiFi. *arXiv preprint arXiv:2301.00250*.

[8] Putterer, F. (2021). Vital Sign Monitoring and Movement Tracking using Commodity-Off-The-Shelf Channel State Information. *Technische Universität München*. https://doi.org/10.13140/RG.2.2.30383.82081

[9] Xie, Y., Li, Z., & Li, M. (2015). Precise Power Delay Profiling with Commodity WiFi. *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 53-64. https://doi.org/10.1145/2789168.2790124

[10] Espressif. (s.f.). ESP-CSI [Consultado el 18 de agosto de 2024]. https://github.com/espressif/esp-csi

[11] Marković, M. (2023, noviembre). Why is ESP the most used IoT connectivity MCU [Consultado el 18 de agosto de 2024]. https://www.byte-lab.com/why-is-esp-the-most-used-iot-connectivity-mcu/

[12] TP-Link. (s.f.). TL-WDR4300 Router Gigabit Inalámbrico de Banda Dual N750 [Consultado el 27 de julio de 2024]. https://www.tp-link.com/es/home-networking/wifi-router/tl-wdr4300/

[13] Xie, Y., Li, Z., & Li, M. (s.f.). Atheros CSI Supported Devices List [Consultado el 27 de julio de 2024]. https://wands.sg/research/wifi/AtherosCSI/DeviceList.html

[14] Xie, Y. (2015a). Atheros CSI Tool [Consultado el 09 de julio de 2024]. https://github.com/xieyaxiongfly/Atheros-CSI-Tool

[15] Xie, Y. (2015b). Atheros CSI Tool UserSpace APP. https://github.com/xieyaxiongfly/Atheros-CSI-Tool-UserSpace-APP

[16] Xie, Y., & Li, M. (2015). Atheros CSI Tool User Guide [Recuperado de GitHub]. https://wands.sg/research/wifi/AtherosCSI/document/Atheros-CSI-Tool-User-Guide.pdf

[17] Colaboradores de Wikipedia. (2024, julio). Software-defined radio [Consultado el 02 de septiembre de 2024]. https://en.wikipedia.org/w/index.php?title=Software-defined_radio&oldid=1237722586

[18] linux-80211n-csitool. (2014). https://github.com/dhalperi/linux-80211n-csitool/

[19] Halperin, D., Hu, W., Sheth, A., & Wetherall, D. (2011a). Tool Release: Gathering 802.11n Traces with Channel State Information. *ACM SIGCOMM Comput. Commun. Rev.*, *41*(1), 53.

[20] Halperin, D., Hu, W., Sheth, A., & Wetherall, D. (2011b). Tool release: gathering 802.11n traces with channel state information. *SIGCOMM Comput. Commun. Rev.*, *41*(1), 53. https://doi.org/10.1145/1925861.1925870

[21] Alonso, R. (s.f.). En el primer trimestre de 2021, los mineros compraron más de 700.000 GPUs [Consultado el 02 de marzo de 2025]. https://hardzone.es/noticias/tarjetas-graficas/ventas-gpu-mineria/

[22] Schneider, J., & Smalley, I. (s.f.). CPU vs. GPU for machine learning [Consultado el 02 de marzo de 2025]. https://www.ibm.com/think/topics/cpu-vs-gpu-machine-learning

[23] Güler, R. A., Neverova, N., & Kokkinos, I. (2018). DensePose: Dense Human Pose Estimation In The Wild. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*.

[24] Schulz, M., Wegemer, D., & Hollick, M. (2017). Nexmon: The C-based Firmware Patching Framework. https://nexmon.org

[25] Gringoli, F., Schulz, M., Link, J., & Hollick, M. (2019). Free Your CSI: A Channel State Information Extraction Platform For Modern Wi-Fi Chipsets. *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization (WiNTECH '19)*, 21-28. https://doi.org/10.1145/3349623.3355477

[26] Hernandez, S. M., & Bulut, E. (2022). WiFi Sensing on the Edge: Signal Processing Techniques and Challenges for Real-World Systems. *IEEE Communications Surveys and Tutorials*, *20*(10), 1-1. https://doi.org/10.1109/COMST.2022.3209144

[27] Hernandez, S. M., & Bulut, E. (2020). Lightweight and Standalone IoT Based WiFi Sensing for Active Repositioning and Mobility. *Proc. 21st International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*.

[28] Hernandez, S. M. (2023a). Android App for Labelling CSI Collected Through the ESP32 CSI Toolkit [Consultado el 30 de marzo de 2025]. https://github.com/StevenMHernandez/Android-CSI-Labelling-App

[29] Hernandez, S. M. (2023b). Android Library for Receiving CSI Data Through Serial from an ESP32 [Consultado el 30 de marzo de 2025]. https://github.com/StevenMHernandez/Android-ESP32-CSI

[30] fpeci. (2021, junio). csi length 0 [Repositorio Atheros_CSI_tool_OpenWRT_src]. https://github.com/xieyaxiongfly/Atheros_CSI_tool_OpenWRT_src/issues/62#issuecomment-2016866173

[31] sayaz. (2018, abril). Capturing of HT sounding packets [Repositorio Atheros-CSI-Tool]. https://github.com/xieyaxiongfly/Atheros-CSI-Tool/issues/24

[32] Yang, Z., Zhang, Y., Chi, G., & Zhang, G. (2024). Hands-on Wireless Sensing with Wi-Fi: A Tutorial. Understanding CSI [Consultado el 11 de julio de 2024]. http://tns.thss.tsinghua.edu.cn/wst/docs/pre/

[33] Community, S. (2018). How channel state information is calculated from Sounding Packet [Consultado el 11 de julio de 2024]. https://dsp.stackexchange.com/questions/47187/how-channel-state-information-is-calculated-from-sounding-packet

[34] Slobogin, C. (2003). Public Privacy: Camera Surveillance of Public Places And The Right to Anonymity. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.364600

[35] Biddle, S. (2019). *Amazon's Ring Is a Perfect Storm of Privacy Threats* [Consultado el 15 de abril de 2025]. The Intercept. https://theintercept.com/2019/01/10/amazon-ring-security-camera/

[36] Ackerman, E. (2020). *Ring's Indoor Security Drone Patrols Your Home While You're Away* [Consultado el 15 de abril de 2025]. IEEE Spectrum. https://spectrum.ieee.org/ring-indoor-security-drone

[37] Frey, C. (2016). *Revealed: how facial recognition has invaded shops – and your privacy* [Consultado el 14 de abril de 2025]. The Guardian. https://www.theguardian.com/cities/2016/mar/03/revealed-facial-recognition-software-infiltrating-cities-saks-toronto

[38] Fowler, G. A. (2019). *Privacy battle brews over home security cameras* [Consultado el 16 de abril de 2025]. The Seattle Times. https://www.seattletimes.com/business/privacy-battle-brews-over-home-security-cameras/

[39] Véliz, C. (2020). *Privacidad es poder: Datos, vigilancia y libertad en la era digital*. Penguin Random House Grupo Editorial.

Apéndices

A

Apéndice I

A.1. Conceptos teóricos

Frecuencia

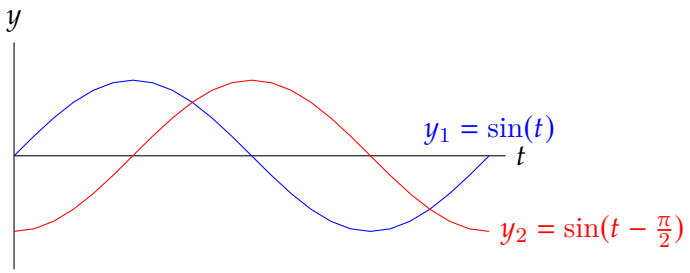
La frecuencia de una onda es el número de ciclos completos que realiza la onda en una unidad de tiempo. La frecuencia está relacionada con la rapidez con la que la señal oscila, lo que influye en la velocidad de transmisión de datos.

Longitud de onda

La longitud de onda es la distancia entre dos puntos adyacentes en la misma fase de una onda. En el caso de las señales WiFi, la longitud de onda está determinada por la frecuencia de la señal y el medio a través del cual se propaga.

Fase

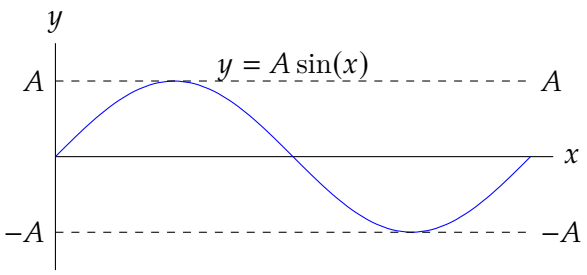
La fase de una onda se refiere a la posición relativa de la onda en su ciclo. Es decir, indica en qué punto de su ciclo se encuentra la onda en un momento dado. Como este es un concepto un tanto abstracto, pero muy importante, pongamos un ejemplo: imagina una onda sinusoidal que se repite a lo largo del tiempo. La fase describe cuánto ha avanzado la onda a lo largo de su ciclo en comparación con un punto de referencia. Por ejemplo, si dos ondas tienen la misma frecuencia pero están desfasadas en 90 grados, significa que una onda ha avanzado un cuarto de ciclo con respecto a la otra.



En esta representación, podemos ver como la onda $y_2 = sin(t - \frac{\pi}{2})$ tiene un desfase de $\frac{\pi}{2} rad$ en comparación con la onda y_1. Lo que significa que y_2 está adelantada respecto a la onda y_1 por un cuarto de ciclo.

Amplitud

La amplitud de una onda se refiere a la altura máxima o el valor máximo de la oscilación de la onda desde su posición de equilibrio. En el caso de ondas electromagnéticas, como las señales WiFi, la amplitud está relacionada con la intensidad de la señal.



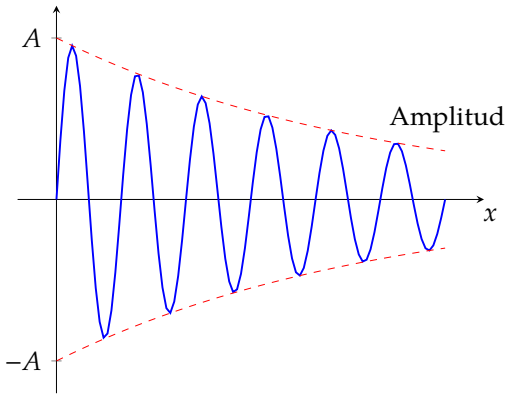
En este ejemplo, la amplitud se representa con la letra A.

Atenuación

La atenuación se define como la disminución progresiva de la intensidad o potencia de una señal (ya sea una onda electromagnética, acústica o de cualquier otra naturaleza) a medida que esta se propaga a través de un medio. Matemáticamente se puede representar como una función logarítmica:

$$A = 10 \log \left(\frac{I_1}{I_0} \right)$$

donde I_0 es la potencia inicial de la señal e I_1 es la potencia después de propagarse por el medio.



A.2. Cálculo del CSI

Para describir matemáticamente la trayectoria que sigue una señal Wi-Fi al propagarse por un canal es necesario tener en cuenta el entorno —si es al aire libre, si es un entorno

doméstico con la presencia de objetos y paredes que pueden funcionar como pantallas que modifiquen la trayectoria de las señales, etc.—. Así pues, tenemos dos modelos aplicables en diferentes escenarios [32], [33]:

A.2.1. Propagación por trazado de rayos

El modelo de trazado de rayos se utiliza para modelar la propagación de señales en entornos interiores. En estos entornos, una señal enviada por el transmisor llega al receptor a través de múltiples trayectorias debido a la reflexión de las ondas de radio. Cada una de estas trayectorias experimenta atenuación y un cambio de fase. La señal recibida es una superposición de estas versiones modificadas de la señal transmitida.

Este modelo establece que la fuerza de la señal recibida en la base compleja puede expresarse como la suma de las componentes de múltiples trayectorias, cada una con su propia amplitud y fase. En consecuencia, el indicador de fuerza de la señal recibida (RSSI) puede expresarse en términos de la potencia de la señal recibida.

El CIR (respuesta al impulso del canal) y el CFR (respuesta en frecuencia del canal) son herramientas clave para caracterizar este fenómeno de trayectorias múltiples, ya que ofrecen una representación detallada de cómo las señales se propagan y se ven afectadas en el espacio de transmisión. Aunque el modelo de *ray-tracing* se utiliza ampliamente en aplicaciones de localización y seguimiento inalámbrico, su principal desventaja es que se basa en suposiciones simplificadas del entorno. En entornos complejos, donde las señales sufren difracción o interferencias, el modelo puede tener dificultades para recuperar toda la información espacial a partir de datos limitados del CSI.

Modelo de Canal Basado en Multipath

La señal recibida V en base compleja es la suma de N componentes de trayectorias múltiples:

$$V = \sum_{n=1}^{N} |V_n| e^{-j\phi_n} \tag{A.1}$$

donde V_n es la amplitud (atenuación) de la n-ésima trayectoria y ϕ_n es la fase, que depende del retardo de la trayectoria.

Relación con la Geometría del Entorno

Cada trayectoria puede modelarse geométricamente por su distancia de propagación d_n tal que

$$\phi_n = \frac{2\pi d_n}{\lambda} \tag{A.2}$$

donde λ es la longitud de onda. La atenuación $|V_n|$ incluye pérdidas por trayectoria directa y pérdidas por reflexión en superficies —según los coeficientes de reflexión de

cada superficie—. De manera que,

$$|V_n| = \frac{G_t G_r}{\left(\frac{4\pi d_n}{\lambda}\right)^2} \cdot \prod_{m=1}^{M_n} \Gamma_m \tag{A.3}$$

donde G_t y G_r son las ganancias de las antenas Tx y Rx, Γ_m es el coeficiente de reflexión en el m-ésimo punto de rebote y M_n es el número de reflexiones en la n-ésima trayectoria.

Respuesta al Impulso del Canal (CIR)

Bajo el modelo de trazado de rayos, la CIR, $h(t)$, es:

$$h(t) = \sum_{n=1}^{N} \alpha_n e^{-j\phi_n \delta(t-\tau_n)} \tag{A.4}$$

donde α_n es la atenuación compleja —incluye pérdidas y fases de reflexión—. $\tau_n = \frac{d_n}{c}$ es el retardo asociado a la trayectoria n, con c la velocidad de la luz en el vacío.

Respuesta en Frecuencia del Canal (CFR)

La transformación de $h(t)$ al dominio de la frecuencia da la respuesta en frecuencia $H(f)$:

$$H(f) = \sum_{n=1}^{N} \alpha_n e^{-j(2\pi f \tau_n + \phi_n)} \tag{A.5}$$

Cada término representa una trayectoria que introduce una atenuación y un retardo dependientes de la frecuencia, causando el fenómeno de desvanecimiento selectivo en frecuencia.

No obstante, a pesar de que este modelo matemático permite teorizar la extracción de distancias d_n y retardos τ_n a partir del CFR, en la práctica las trayectorias se mezclan de forma no trivial cuando hay difracción, dispersión o ruido. Además, la resolución temporal y espacial está limitada por el ancho de banda del sistema. Esta es la razónm por la que muchos sistemas actuales combinan el modelo físco con técnicas estadísticas o de *machine learning*.

A.2.2. Modelo de Dispersión (*Scattering Model*)

Lo más común es que dentro de un entorno cerrado, el rumbo de la señal no tome un camino en línea recta hacia el dispositivo en cuestión, sino que la señal puede rebotar por diferentes obstáculos. Por ende, este modelo trata a todos los objetos del entorno como dispersores que difunden las señales en todas las direcciones.

$$H(f,t) = \sum_{n=1}^{N} \alpha_n(f,t) e^{-j\phi_n(f,t)} \tag{A.6}$$

Donde N es el número total de caminos de reflexión, $\alpha_n(f,t)$ es la atenuación de amplitud y $\phi(f,t)$ es la fase correspondiente. Similar al modelo de trazado de rayos. Existen otros modelos para diferentes casos. De todos modos, creo que dentro de los entornos domésticos o clases del instituto, este el modelo que mejor explica la propagación de las señales.
Volviendo a los pasos en el cálculo del CSI, uno de los métodos conocidos es: transmitiendo una Señal Piloto (también conocida como paquete de sondeo o HT Sounding Packet), el cuál incluye una flag. Esta flag es predefinida y se conoce tanto en el transmisor como en el receptor. El receptor capta la señal piloto que ha atravesado el canal y debido a las características del mismo, la flag recibida será una versión modificada de la señal piloto original. El receptor comparará la señal recibida con la original y determinará cómo el canal la ha alterado.
Usando la comparación, se calcula la respuesta del canal que es representada por un valor complejo $H(f)$, que indica tanto la ganancia (amplitud) como el desfase introducido por el canal. De modo que podemos modelar la señal recibida $Y(f)$ como:

$$Y(f) = H(f) \cdot X(f) + N(f) \tag{A.7}$$

Donde, en función de la frecuencia f, $H(f)$ es la respuesta del canal, $X(f)$ es la señal piloto y $N(f)$ es el ruido presente.
Para calcular $H(f)$, procedemos a despejar la función de la ecuación A.7:

$$H(f) = \frac{Y(f) - N(f)}{X(f)} \tag{A.8}$$

El valor complejo $H(f)$ lo podemos descomponer en su parte real a —indica la fase de la señal— y en su parte compleja b —indica la componente en cuadratura de la señal—. Geométricamente, la magnitud $|H(f)|$ y la fase $\angle H(f)$ se pueden calcular como:

- Amplitud:

$$A = |H(f)| = \sqrt{a^2 + b^2} \tag{A.9}$$

- Fase:

$$\theta = \angle H(f) = \arctan\left(\frac{b}{a}\right) \tag{A.10}$$

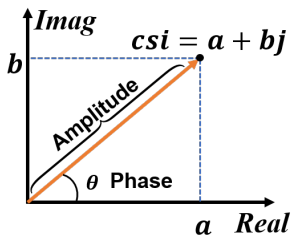


Figura A.1: *Representación gráfica de la parte compleja y real del CSI. Imagen extraída de la web Wands del proyecto Atheros CSI Tool* [9].