

# Proyecto Fin de Grado GITT

Intensificación en Sistemas Electrónicos

## Caracterización de dispositivos Wi-Fi comerciales para su aplicación en comunicaciones entre UAVs y dispositivos de tierra

Autor: Carlos Beltrán Hortelano Sancho

Tutor: Daniel Gutiérrez Reina

**Dept. de Ingeniería Electrónica  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla**

Sevilla, 2017





Proyecto Fin de Grado  
GITT  
Intensificación en Sistemas Electrónicos

Caracterización de dispositivos Wi-Fi comerciales para  
su aplicación en comunicaciones entre UAVs y  
dispositivos de tierra

Autor:  
Carlos Beltrán Hortelano Sancho

Tutor:  
Daniel Gutiérrez Reina  
Doctor Ingeniero en Electrónica

Dept. de Ingeniería Electrónica  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2017

Trabajo Fin de Grado: Caracterización de dispositivos Wi-Fi comerciales para su aplicación en comunicaciones entre UAVs y dispositivos de tierra

Autor: Carlos Beltrán Hortelano Sancho

Tutor: Daniel Gutiérrez Reina

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2017

El Secretario del Tribunal

*A todos los que lo hicieron posible*



# Agradecimientos

---

Adaptando uno de los refranes más populares en lengua castellana, escribió Ángel de Saavedra, duque de Rivas:

*Porque el ser agradecido  
la mayor obligación es  
para el hombre bien nacido*

Las escuetas líneas que siguen no son más que un intento de acreditar dicha condición.

Es obligado que la primera mención vaya dirigida al tutor de este proyecto, Daniel Gutiérrez. Desde el día de octubre de 2016 en el que le mostré mi interés por este trabajo su disponibilidad, atención y diligencia han estado siempre presentes. Gracias a su planificación, acatada por quien escribe estas líneas, este proyecto se ha realizado con paciencia y constancia. Sus consejos e indicaciones han resultado indispensables.

La colaboración de Sergio Vigorra fue inestimable. Su Trabajo de Fin de Máster fue el germen de este, y no sólo eso: el dron que él fabricó fue el que se utilizó para realizar las pruebas de comunicaciones en vuelo. A él debo agradecerle su tiempo, proactividad durante la fase de obtención de resultados, interés y algunos comentarios que a él pasaron desapercibidos cuando los pronunció y que a mí, sin embargo, me ayudaron a mejorar este trabajo.

En lo que corresponde a la redacción de este proyecto, Salvador Núñez ofreció sugerencias de estilo y emprendió una minuciosa búsqueda de erratas, propia de sabuesos. Toda la responsabilidad por las que permanezcan recaen sobre mí.

Por último, este es el fruto de los conocimientos y las buenas prácticas adquiridos gracias a la labor de decenas de profesores durante estos años en la Escuela. Dirijo mi sincero agradecimiento hacia todos ellos.

Todas las personas referidas han sido, de manera puntual o persistente, acompañantes en un camino no exento de obstáculos y del que este proyecto no es más que una marca. Se trata por tanto de un camino que no termina aquí pues, como dijo otro poeta español, se hace al andar.

Y sigo andando.

*Carlos Beltrán Hortelano Sancho*

*Sevilla – Cádiz – Sevilla, septiembre de 2017*





# Resumen

---

Los vehículos aéreos no tripulados (popularmente conocidos como drones) constituyen en la actualidad una tecnología emergente y con aplicaciones en múltiples campos. Este trabajo pretende construir un sistema de comunicaciones inalámbrico basado en el estándar Wi-Fi entre dos Raspberry Pi para su uso en estos dispositivos y medir su calidad. Antes de conseguir este objetivo será necesario cumplir varios hitos:

- Preparar las Raspberry Pi para que se adapten a las exigencias del sistema de comunicaciones.
- Considerar las opciones disponibles en el mercado para conseguir la comunicación inalámbrica.
- Determinar qué características de las comunicaciones quieren medirse y aplicar las herramientas necesarias para ello.
- Diseñar unos programas que consigan obtener, de las herramientas mencionadas en el punto anterior, los parámetros para definir la calidad de las comunicaciones.

Una vez procesados los parámetros de calidad de las comunicaciones se podrá reflexionar acerca de las posibilidades de un sistema de comunicaciones inalámbrico basado en Raspberry Pi, y proponer aquellas aplicaciones que, de acuerdo con los datos obtenidos y los recursos manejados, resultaran más provechosas para su uso en UAVs.



# Abstract

---

Unmanned aerial vehicles (popularly known as drones) are currently an emerging technology with applications in multiple fields. This paper aims to build a wireless communications system based on the Wi-Fi standard between two Raspberry Pi for their use in these devices and measure their quality. Before achieving this goal, several milestones should be met:

- Prepare the tools so they adapt to the demands of the communications system.
- Consider the options available in the market for wireless communication.
- Determine what characteristics of the communications want to be measured and apply the tools needed to do so.
- Design programs able to obtain, from the tools mentioned in the previous point, the parameters to define the quality of communications.

Once the quality parameters of the communications are processed, it will be possible to reflect on the possibilities of a wireless communications system based on Raspberry Pi, and to propose those applications that, according to the data obtained and the resources handled, will be more profitable for its use in UAVs.

# Índice

Agradecimientos	vii
Resumen	ix
Abstract	xi
Índice	12
Índice de abreviaturas, siglas y acrónimos	13
<b>1 Objetivos y motivación del proyecto</b>	<b>15</b>
<b>2 Introducción: wi-fi para comunicaciones entre UAVs y dispositivos de tierra</b>	<b>16</b>
2.1 Antecedentes de los vehículos aéreos no tripulados	16
2.2 Introducción a IEEE 802.11	17
2.3 Dispositivos comerciales	21
2.4 Trabajos previos	22
<b>3 Hardware y software utilizados</b>	<b>25</b>
3.1 Raspberry Pi	25
3.1.1. Instalación de la imagen Raspbian, actualización de software y ajustes menores	29
3.1.2. Manejo de Raspberry Pi desde PC y conexión a la red	29
3.1.3. Despliegue de la red ad-hoc	31
3.1.4. Sincronización de fecha y hora: el protocolo NTP	33
3.2 Alimentación	35
3.3 Dispositivos de conexión inalámbrica	36
3.4 Elementos software	37
3.3.1. Lenguaje de programación: Python	37
3.3.2. Entorno de desarrollo: Anaconda	37
3.3.3. Programa cliente-servidor basado en TCP y UDP	37
3.3.4. Wireshark: gestión de capturas de tráfico	39
3.3.5. Programas de procesamiento de datos relativos a las características de las comunicaciones inalámbricas	42
3.3.6. Ejecución de tareas: SSH	43
<b>4 Banco de pruebas</b>	<b>45</b>
4.1 Pruebas E.1 y E.2: UDP y TCP con dongle	47
4.2 Pruebas E.3 y E.4: UDP y TCP con transmisor integrado	50
4.3 Pruebas D.1 y D.2: UDP y TCP con dongle	52
4.4 Pruebas D.3 y D.4: UDP y TCP con transmisor integrado	56
<b>5 Análisis de resultados y futuros trabajos</b>	<b>60</b>
Referencias	62
Índice de Figuras	65
Índice de Tablas	67
Códigos utilizados	69
Mensaje transmitido	76

# Índice de abreviaturas, siglas y acrónimos

---

AP	Access Point
BLOS	Beyond Line of Sight
CDMA	Code-division Multiple Access
CPU	Central Processing Unit
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSV	Comma Separated Value
EMI	Electromagnetic Interference
FTP	File Transfer Protocol
GCS	Ground Control Station
GPS	Global Positioning System
GUI	Graphical User Interface
GPIO	General Purpose Input Output
GPU	Graphics Processor Unit
HLL	High Level Language
HTTP	HyperText Transfer Protocol
IBSS	Independent Basic Service Set
IEEE	Institute of Electrical and Electronic Engineers
ISI	Intersymbol Interference
ISM	Industrial, Scientific and Medical
ITU	International Telecommunications Union
LOS	Line of Sight
LSF	Large-scale Fading
MANET	Mobile Ad-hoc Network
NTP	Network Time Protocol
OSI	Open Systems Interconnection
PRR	Packet Reception Rate
RAM	Random Access Memory
RIP	Routing Information Protocol
RMC	Resource Monitoring and Control
RPi	Raspberry Pi
RTC	Real-Time Clock
RTT	Round-trip Time
RX	Reception
SCP	Secure Copy Protocol
SSH	Secure Shell
SMTP	Simple Mail Transfer Protocol
SSF	Small-scale Fading

TCP	Transmission Control Protocol
TX	Transmission
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
USB	Universal Serial Bus
USD	United States Dollar
WLAN	Wireless Local Area Network

# 1 OBJETIVOS Y MOTIVACIÓN DEL PROYECTO

---

*One accurate measurement is worth a thousand expert opinions.*

*- Grace Hopper, matemática y científica de la computación -*

Este trabajo persigue ampliar el conocimiento relativo a las comunicaciones que pueden efectuarse entre los dispositivos de nueva generación Raspberry Pi. Las peculiaridades de este producto, hacen que el uso de estos dispositivos sea una opción a considerar en proyectos que precisen de sistemas de comunicaciones económicos, de reducido tamaño y livianos. Para comprobar la calidad de dichas transmisiones se probarán las Raspberry Pi en escenarios tanto estáticos como dinámicos, siendo estos últimos los que, en caso de devolver resultados satisfactorios, pueden dar lugar a un mayor número de aplicaciones. En concreto, las pruebas dinámicas evaluarán la calidad de las comunicaciones una estación de control en tierra (o GCS, por sus siglas en inglés) y un UAV. Se calcularán parámetros tales como tasa binaria, radio de cobertura, tasa de recepción de paquetes y retardo medio, valiéndose del estándar 802.11 y modo de conexión ad-hoc de acuerdo con la funcionalidad del hardware disponible. Para propiciar las comunicaciones entre la estación de tierra y el UAV se contará con uno de los muchos *dongles* disponibles en el mercado. Debido a las distintas características de estos dispositivos será necesario un estudio pormenorizado que determine cuál se adapta mejor a los requerimientos del proyecto. Además, se aprovechará la incorporación de un módulo Wi-Fi integrado en las Raspberry Pi para así evaluar las diferencias entre ambas interfaces.

Un aspecto sobre el que habrá que realizar especial hincapié al tratar las comunicaciones donde se ven implicados UAVs es que estos trabajan en un espacio tridimensional. Esto supone un desafío, ya que se trata de una coyuntura muy distinta a la de la transmisión de datos en espacios de dos dimensiones, incluyendo bajo este concepto a aquellas zonas tridimensionales que pueden modelarse como planarias sin que la aproximación perjudique sensiblemente al rendimiento de las comunicaciones. Las características de los medios submarinos, las distintas capas de la atmósfera o el espacio exterior influyen palmariamente en la naturaleza del medio de transmisión y, por ende, precisan de modelos particularizados para cada caso y que distan mucho de los que se utilizan en medios bidimensionales convencionales.

En definitiva, se pretende la caracterización de las conexiones Wi-Fi entre dispositivos Raspberry Pi puestos a prueba en distintos escenarios y, a partir de los datos generados y procesados haciendo uso de herramientas como Wireshark y programas escritos en lenguaje Python, obtener conclusiones relativas al rendimiento de las comunicaciones efectuadas.

# 2 INTRODUCCIÓN: WI-FI PARA COMUNICACIONES ENTRE UAVs Y DISPOSITIVOS DE TIERRA

---

*If we all worked on the assumption that what is accepted as true is really true, there would be little hope of advance.*

*- Orville Wright, padre de la aviación -*

El presente proyecto consiste en el despliegue de una red inalámbrica ad-hoc y la medición de la calidad de comunicaciones con base en criterios estandarizados. Para ello se contará con dos Raspberry Pi 3 Model B ®. Las configuraciones de redes a probar se describirán de forma prolija en sucesivas páginas, pero a grandes rasgos consistirán en dos Raspberry Pi enviándose mensajes en un entorno estático y, posteriormente, en una de ellas dispuesta sobre un vehículo aéreo no tripulado (UAV, por sus siglas en inglés) y la restante en una estación de tierra. Los sistemas transmisor y receptor los constituirán dos con la imagen Jessie del sistema operativo Raspbian ® corriendo en ellas. Insertas en puertos USB de cada una de las Raspberry Pi se encuentra un módulo Wi-Fi para conseguir la conectividad entre ambas a partir de distintas configuraciones que serán programadas para tabular sus parámetros de servicio, si bien se prescindirá de estos dispositivos cuando se mida la calidad de la interfaz Wi-Fi interna con la que cuenta el modelo 3B. Para monitorizar las comunicaciones Wi-Fi y evaluar su calidad, en la estación de tierra se contará con el analizador de protocolos Wireshark ®, que habrá sido previamente instalado en la imagen de Raspbian ®. Será imprescindible considerar aspectos cruciales en la comunicación, tales como las limitaciones que ofrece el estándar 802.11 con nodos desplazándose tridimensionalmente y, además, en espacios abiertos donde pueden verse expuestos a una cantidad inconmensurable de otras señales radioeléctricas (ruido), además de obstáculos físicos.

## 2.1 Antecedentes de los vehículos aéreos no tripulados

Al igual que muchos descubrimientos e invenciones realizados a lo largo de la historia, la primera motivación de los vehículos no tripulados fue militar. Sus orígenes se remontan a los primeros tiempos de nuestra era, mucho antes del primer vuelo a motor en 1903 con uno de los hermanos Wright a los mandos.

En el siglo III d.C. el estratega chino Zhuge Liang ideó unos globos de papel que en su interior contenían una llama que, al mismo tiempo que calentaban el aire y propiciaban que el globo se alzase, suponían una amenaza contra los enemigos que, al ver llamas de fuego de origen ignoto suspendidas en el aire, las consideraban una amenaza divina. Muchos años después, en 1848, los conocimientos científicos dispararon de forma considerable esos miedos atávicos, pero los sistemas para amedrentar a los pueblos hostiles se mantuvieron básicamente imperturbables. En dicho año, Austria ejecutó un ataque aéreo sobre Venecia por medio de globos propulsados por el viento que portaban explosivos. Posteriormente, tanto la primera como la segunda guerra mundial fueron acicates para el perfeccionamiento de estos vehículos. En la actualidad la tecnología en la que se basan los drones se encuentra en constante evolución y ello, unido a la universalización de su uso debido a sus precios cada vez más asequibles, variedad de tamaños y pesos y abanico de prestaciones (ver Tabla 1), da lugar a un amplio rango de aplicaciones, desde las más intuitivas, como la toma de imágenes aéreas con fines recreativos o de vigilancia, hasta las más sofisticadas. Bajo este último calificativo se engloban actividades como la meteorología, la agricultura, el transporte de mercancías, la investigación de zonas de acceso difícil o arriesgado para las personas, tales como regiones volcánicas, misiones humanitarias u operaciones de rescate. En todas estas misiones los drones brindan la capacidad de monitorizar un determinado entorno y, con los datos que éste aporta, tomar de forma rápida una decisión que ayude a completarla de forma exitosa.



Categoría	Peso al despegue (kg)	Altitud de operación (m)	Velocidad (km/h)
Grupo 1	< 9	< 366 sobre el nivel de la tierra	< 185
Grupo 2	9,55 - 25	< 1588 sobre el nivel de la tierra	< 463
Grupo 3	< 599	< 8164 sobre el nivel del mar	
Grupo 4	> 599		Cualquiera
Grupo 5		> 8164 sobre el nivel del mar	

Tabla 1. Clasificación de los vehículos aéreos no tripulados

En paralelo, la proliferación masiva de estas aeronaves ha generado problemas no sólo técnicos, sino también relacionados con la seguridad aérea o la privacidad. Numerosos gobiernos nacionales y supranacionales han elaborado o están en proceso de desarrollar normativas encaminadas a garantizar el correcto uso de estos dispositivos. Quizá esta sobrevenida urgencia por regular el tráfico de los UAVs sea el síntoma más palpable de que estos dispositivos han llegado para quedarse<sup>1</sup>.

Como se deduce de las aplicaciones mencionadas anteriormente, todas ellas precisan en menor o mayor medida de la transmisión y recepción de datos para su correcto funcionamiento. Así, los drones utilizados en el ámbito del transporte necesitan tomar datos de satélites que les indiquen la ruta idónea hasta su destino y notificar al lugar desde donde partió cualquier incidencia que se produzca durante el trayecto. Del mismo modo, los que sobrevuelan una zona catastrófica, como un bosque asolado por un incendio, deben adquirir y transmitir imágenes y otros datos (temperatura, dirección y velocidad del viento, etc.) al contingente de bomberos para que éste conozca a qué condiciones se enfrenta a la hora de intentar sofocarlo.

Por la naturaleza móvil de los vehículos aéreos no tripulados la transmisión de datos a través de un medio cableado debe ser directamente descartada. No tendría sentido que un vuelo se viese limitado por la longitud del cable que une al dron con la estación de tierra: al final, se trataría de un sistema análogo a una cometa, con la salvedad de que entre sus extremos se produciría un intercambio de datos. Para dotar de mayor libertad de maniobra al vehículo resulta obligado recurrir a redes de comunicaciones basadas en estándares no cableados. Es en este punto cuando resulta pertinente introducir el estándar 802.11.

## 2.2 Introducción a IEEE 802.11

Por medio del estándar 802.11 el IEEE define las comunicaciones basadas en red local inalámbrica o WLAN. Las WLAN responden a la necesidad de proporcionar conectividad en prácticamente cualquier situación a dispositivos portátiles, dando lugar a redes flexibles y versátiles que se caracterizan por su movilidad, su capacidad para llegar a zonas donde es difícil instalar redes cableadas y su coste indiscutiblemente más económico respecto de estas últimas.

Para garantizar las comunicaciones inalámbricas, el estándar IEEE 802.11 utiliza las bandas del espacio radioeléctrico denominadas ISM (Industrial, Scientific and Medical). La marca Wi-Fi certifica la compatibilidad entre las distintas variantes del estándar y establece criterios de calidad de los nodos que conforman la red. Desde su versión prístina de 1997, las investigaciones de los grupos de trabajo del IEEE han ido encaminadas a mejorar las prestaciones del estándar, sobre todo en lo concerniente a la tasa máxima de

<sup>1</sup> nytimes.com. (2016). *Drone Regulations Should Focus on Safety and Privacy*. [online] Disponible en: [https://www.nytimes.com/2016/01/10/opinion/sunday/drone-regulations-should-focus-on-safety-and-privacy.html?\\_r=0](https://www.nytimes.com/2016/01/10/opinion/sunday/drone-regulations-should-focus-on-safety-and-privacy.html?_r=0) [Accedido el 6 de mayo de 2017]. Este artículo, partiendo de la constancia del creciente número de drones en manos de particulares, aborda problemas de seguridad y privacidad relacionados con su uso y las normativas elaboradas para paliarlos.

transmisión. La Tabla 2 muestra la evolución del estándar.

Estándar	Lanzamiento	Frecuencia (GHz)	Ancho de banda (MHz)	Tasa de transmisión máxima (Mb/s)	Modulación	Rango exterior (m)
802.11-1997	1997	2.4	20	2	DSSS, FHSS	100
802.11a	1999	5	20	54	OFDM	120
802.11b	1999	2.4	20	11	DSSS	140
802.11g	2003	2.4	20	54	DSSS, OFDM	140
802.11n	2009	2.4, 5	20, 40	600	OFDM	250
802.11ac	2013	5	40, 80, 160	6930	OFDM	300
802.11ad	2016	60	2160	6760	SC, OFDM	100

Tabla 2. Estándares 802.11

En la actualidad prácticamente todos los estándares trabajan en las bandas de 2.4 y 5 GHz del espectro radioeléctrico. En función de la naturaleza del medio en el que se desarrollen las comunicaciones inalámbricas, una banda u otra resulta preferible. En la Tabla 3 se explica de manera sucinta la diferencia entre ambas.

Banda (GHz)	Canales	Rango de cobertura	Presencia de interferencias
2.4	3, no solapados	Amplio	Mayor
5	23, no solapados	Reducido	Menor

Tabla 3. Comparación de prestaciones entre bandas de 2.4 y 5 GHz

En comunicaciones que impliquen grandes distancias entre nodos la banda de 2.4 GHz es preferible. Se recomienda su uso para tráfico elástico, es decir, aquel que puede adaptarse a variaciones en el medio de transmisión sin perjudicar a la aplicación. Se trata de una banda más proclive a interferencias debido a que se comparte esta franja del espectro radioeléctrico con tecnologías de uso masivo como telefonía inalámbrica, aparatos microondas, Bluetooth e, incluso, las alarmas antirrobo de los vehículos a motor.

La banda de 5 GHz, por el contrario, tiene un radio de cobertura menor. En un entorno sin línea de visión (o LOS, por sus siglas en inglés) entre antenas transmisora y receptora, las señales con frecuencias altas se verán atenuadas debido a la presencia de obstáculos que serán salvados por otras con frecuencias más bajas [4]. En cuanto a la presencia de interferencias estas son menores, debido a la ocupación escasa de esta banda en comparación con la de 2.4 GHz. La banda de 5 GHz se destina a tráfico inelástico, esto es, aquel que precisa de un régimen binario elevado y no puede permitirse retrasos en la transmisión.

El estándar 802.11ad, lanzado en 2016, supone una revolución al utilizar la banda alrededor de los 60 GHz. Por la desocupación de esta banda, un gran número de dispositivos pueden trabajar en ella transmitiendo a regímenes binarios muy superiores a los de las bandas convencionales. Se trata de un estándar donde la seguridad se ve reforzada, debido a que resulta complicado interceptar las transmisiones. Sin embargo, y por las razones antes expuestas, el operar en un rango de frecuencias tan elevado del espectro electromagnético provoca pronunciadas atenuaciones que hacen a este estándar poco recomendable en entornos donde haya

elementos que obstaculicen el paso de la señal.

En este proyecto la configuración de IEEE 802.11 a utilizar es ad-hoc. Este modo de conexión resulta propicio en todas aquellas operaciones que involucren drones, ya que no requiere de puntos de acceso: los propios módulos de conexión inalámbricos, montados sobre los UAVs o las estaciones de tierra, pueden comunicarse entre ellos sin estos dispositivos intermediarios. Esto supone una considerable ventaja en situaciones donde sea primordial una rápida intervención de los drones, ya que de forma presta se puede constituir la red y comenzar la comunicación. No se trata, de todos modos, de una bicoca: entre los inconvenientes asociados a una red de este tipo se encuentran la dificultad a la hora de gestionar la red de forma centralizada (todos los nodos de la red ad-hoc se encuentran al mismo nivel) y, asimismo, la imposibilidad de acceder a internet, a no ser que en uno de los dispositivos de la red, con acceso a internet, se permita una conexión compartida. Es por estos motivos que las redes ad-hoc están orientadas a un grupo limitado de usuarios y no resultan eficientes como redes masivas. Para las características de este proyecto, sin embargo, una red ad-hoc resulta idónea.

Hay desventajas inherentes a las comunicaciones inalámbricas con independencia del estándar IEEE 802.11 con el que se esté trabajando o del modo de conexión utilizada, puesto que se deben única y llanamente a la naturaleza del medio de transmisión. Las investigaciones en el campo de las comunicaciones inalámbricas encaminadas a la elaboración de estándares que respondan a nuevas necesidades pueden mitigar, pero en ningún caso eliminar por completo, estas contingencias. En los párrafos siguientes se desarrollarán los cuatro grandes retos [2] que las tecnologías de comunicación inalámbricas deben domeñar.

En primer lugar, considérese una estación transmisora (en adelante, estación TX) y otra receptora (en adelante, estación RX) en la que la señal puede tomar distintos caminos. Cada una de estas rutas se caracteriza por la mayor o menor presencia de obstáculos y otras incidencias que modifican el comportamiento de la señal en sus parámetros típicos. Este fenómeno se conoce como propagación multitrayecto y es causado por fenómenos de reflexión, difracción y dispersión. De este modo, las interferencias entre señales pueden ser destructivas o constructivas en función de la amplitud, la fase y el retraso con el que estas señales llegan a una antena RX en función del camino que hayan seguido. Esto se conoce como desvanecimiento de pequeña escala, o SSF por sus siglas en inglés, y generalmente da lugar a la atenuación de la señal de interés. Estrechamente relacionado con este concepto está el de desvanecimiento de amplia escala, o LSF por sus siglas en inglés. Son dos fenómenos que, obviamente, no son independientes; todo lo contrario, pues se combinan en la mayoría de ocasiones. La diferencia cuantitativa entre LSF y SSF estriba en que en la primera se dan repentinas caídas de la potencia recibida por un dispositivo en movimiento al pasar desde zonas con LOS hasta otras sin visión directa. Esto se conoce como *shadowing*, y es la principal causa del desvanecimiento de amplia escala. En resumen, mientras que SSF genera atenuaciones suaves en la señal, las provocadas por LSF son abruptas. Debido a la naturaleza de este proyecto, en el que se tendrá un UAV desplazándose en las tres dimensiones de un espacio abierto, el fenómeno de desvanecimiento de amplia escala se deberá tener muy en consideración a la hora de medir la calidad de las comunicaciones.

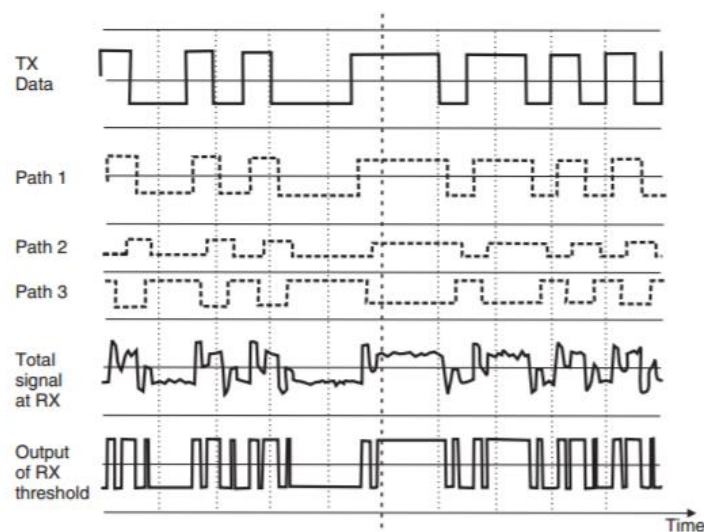


Figura 1. Fenómeno de interferencia intersímbolo

El descrito arriba no es el único problema causado por la propagación multitrayecto. Supóngase que, para una misma señal, cada uno de los caminos por los que puede pasar da lugar a retrasos. Esto da lugar a que en un instante  $t$ , la señal que ha seguido el camino 1 entrega a la antena RX el bit  $k$ , mientras que la señal que llega por el camino 2 entrega el bit  $k-1$ , y por el camino 3 se recibe  $k-2$ . Esto da lugar a la recepción de signos contradictorios en un mismo instante, generando una distorsión de la señal. Este fenómeno se conoce como interferencia intersímbolo, o ISI por sus siglas en inglés. ISI se debe principalmente a los retrasos inherentes a cada trayecto y depende de la proporción entre la duración de cada símbolo y la duración de la respuesta del canal de comunicaciones. La Figura 1 representa esta contingencia. Obsérvese que para los datos transmitidos (TX data), cada camino presenta un retraso distinto. La ruta 1 apenas retrasa respecto del ritmo de los datos emitidos por la antena. La ruta 2, además de un mayor retraso, presenta gran atenuación. En definitiva, los retrasos y debilitamientos de la señal contribuyen a que la secuencia de bits correspondiente a la suma total de las señales no concuerde con los datos transmitidos.

El segundo desafío de las redes inalámbricas se relaciona con algo ya mencionado antes: la existencia de un espacio radioeléctrico compartido con otras tecnologías (algo especialmente notorio en la banda de 2,4 GHz). El espacio radioeléctrico es limitado y, por tanto, preciado. Es por ello que las comunicaciones inalámbricas deben buscar altos índices de eficiencia espectral, es decir, de densidad de tráfico que puede procesarse por unidad de ancho de banda y de área. Para garantizar un reparto eficaz del espectro radioeléctrico que minimice fenómenos de interferencia, la Unión Internacional de Telecomunicaciones (o ITU, por sus siglas en inglés), organismo dependiente de Naciones Unidas, se encarga de asignar regiones del espacio radioeléctrico a distintas tecnologías de comunicaciones inalámbricas en función de sus características y requerimientos. Para más información a este respecto, [3] constituye una referencia fidedigna y completa a pesar de su antigüedad.

El tercer factor que limita de forma decisiva las comunicaciones inalámbricas es el consumo energético. Como se verá en el apartado correspondiente a trabajos previos, son incontables las investigaciones dedicadas a este asunto, y tanto cubren el diagnóstico de la situación (por qué existe una robusta relación entre la capacidad de transmisión y el consumo energético) como exploran posibles soluciones para rebajar la correlación entre ambos factores. De forma resumida, pues no es ésta la principal motivación del proyecto, puede concluirse que las características *deseables* de un sistema de comunicación inalámbrica son las que siguen: se precisa de antenas receptoras con alta sensibilidad (para evitar, por ejemplo, los problemas mentados en párrafos anteriores de este apartado); los dispositivos deben disponer de sistemas de ahorro de energía (la monitorización continua acerca de la situación de la red consume ingentes cantidades de recursos de alimentación) y, al mismo tiempo, los amplificadores con los que trabajan las antenas deben ser altamente eficientes. Por eficiencia de un amplificador se entiende la cantidad de potencia *útil* de salida que se obtiene a partir de potencia de señal a la entrada, y determina el ciclo de trabajo del dispositivo. En este sentido, amplificadores Clase C, con un nivel de eficiencia del 75%, o variaciones del mismo, como Clase F, resultan preferibles (en ningún caso idóneos, pues no existe un amplificador idealmente eficiente).

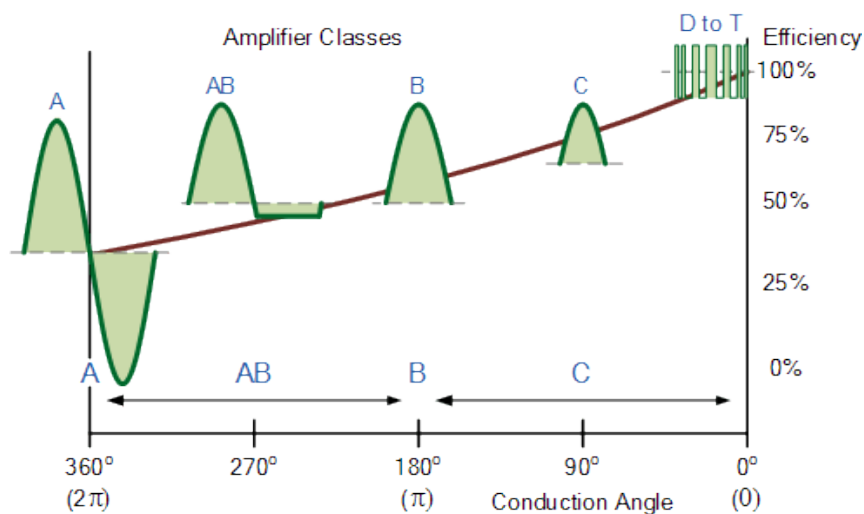


Figura 2. Clasificación de amplificadores según su eficiencia y ángulo de conducción

Sin embargo, una mayor eficiencia de los amplificadores hace de estos menos lineales y por tanto más sensibles a distorsión de la señal. La solución a esta contingencia pasa por modular la señal de acuerdo a configuraciones más sofisticadas (modulaciones con señal envolvente no constante, por ejemplo) que no se vean afectadas por estas no linealidades. Sin embargo, con este tipo de modulaciones se pierde eficiencia espectral, esto es, la capacidad de *acoger* en el sistema de comunicaciones al máximo posible de usuarios [15]. Como se observa, se trata de un círculo vicioso en el que hay que buscar la compensación entre las características referidas.

El último factor con el que deben lidiar las comunicaciones inalámbricas es, posiblemente, el más visible e intuitivo: el de la movilidad. Cuando el usuario (o, en el caso que ocupa este proyecto, un sistema teledirigido) se desplaza, la antena transmisora que le provee de servicio puede cambiar. Es por esta razón que la estación receptora debe enviar con periodicidad tramas al sistema proveedor que informen de su localización y permitan a éste planificar la configuración más adecuada para proporcionar un servicio con unos mínimos requisitos de calidad.

Pese a los inconvenientes comunes a las comunicaciones inalámbricas referidos, lo cierto es que hay diferencias entre las distintas posibilidades que hoy están disponibles en el mercado. Se ha visto, asimismo, que en Wi-Fi se mantiene una lucha entre el rango de cobertura y las interferencias sufridas por la señal. Cuando uno de estos aspectos mejora, el otro empeora: hasta ahora no se ha conseguido una optimalidad simultánea de los mismos. Es por ello que llegados a este punto sería natural preguntarse si otras tecnologías de comunicaciones inalámbricas, véanse ZigBee o Bluetooth, podrían ofrecer mejores prestaciones en lo que a la consecución de las metas de este proyecto se refiere: autonomía, rango de cobertura, caudal, etc.

En el caso de ZigBee, basado en el estándar 802.15-4, su principal inconveniente radica en la pobre tasa de transmisión que ofrece, en ningún caso superior a 250 Kb/s. Dado que esta es una cifra basada en modelos teóricos, lo cierto es que se reduce dramáticamente en la realidad, llegando incluso a una décima parte del valor nominal [20]. Esto se debe principalmente a la naturaleza de CSMA/CA: los numerosos bits llamados *overhead*, es decir, aquellos que no transmiten datos eficaces, representan una considerable proporción del tamaño de la trama y, por lo tanto, del caudal transmitido. A este fenómeno, ya de por sí de gran calado, se suman los propios del medio de transmisión. Es por estas razones que ZigBee es una tecnología apropiada para comunicaciones que no precisan de una gran tasa de transmisión ni demasiado ancho de banda, por lo que se orienta principalmente a control y seguimiento en entornos industriales.

## 2.3 Dispositivos comerciales

La Raspberry Pi será el dispositivo que controlará y procesará las comunicaciones inalámbricas entre la estación de tierra y el vehículo aéreo. Este producto cuenta con un adaptador integrado destinado a las comunicaciones inalámbricas Wi-Fi, no obstante, antes de recurrir a este módulo, las comunicaciones basadas en el estándar IEEE 802.11 se llevarán a cabo a través de un adaptador USB o *dongle*. El modelo de Raspberry Pi dispone de cuatro puertos USB por lo que dedicar uno a las comunicaciones inalámbricas no supone demasiada pérdida de funcionalidad. A la hora de decantarse por uno u otro hay que tener en cuenta características tales como las versiones del estándar que soporta, la compatibilidad con sistemas operativos o si precisa de software adicional a instalar para su funcionamiento, algo que puede mermar el rendimiento de la Raspberry Pi. El mercado ofrece distintas alternativas que responden a esta pretensión, algunos de los cuales se detallan en la Tabla 4.

Fabricante	Modelo	Compatible con Raspbian	Frec. (MHz)	Interfaz	Modos de conexión	Estándares de seguridad
AirLink	AirLink101 AWLL6075 Wireless N Mini USB Adapter	Sí	2.4	b/g/n		64/128-bit WEP, WPA2-PSK, WPA-PSK
Asus	USB-N10	Sí (imagen Wheezy)	2.4	b/g/n		64/128-bit WEP, WPA2-PSK,

						WPA-PSK
<b>Broadcom</b>	Wi-Fi Adapter and 2 Port USB Hub for Raspberry Pi	Sí	2.4	b/g/n	Infraestructura, Wi-Fi Direct, soft-AP	WPA, WPA2, AES/TKIP, WAPI
<b>Edimax</b>	EW-7811Un	Sí	2.4	b/g/n		64/128-bit WEP, WPA2-PSK, WPA-PSK, WPS-compatible
<b>Element14</b>	Wi-Fi	Sí	2.4	b/g/n	Ad-hoc	64/128-bit WEP, WPA/WPA2, WPA-PSK, WPA2-PSK
<b>Iogear</b>	Compact Wireless-N USB Adapter	No	2.4	b/g/n	Ad Hoc, Infraestructura	WEP, TKIP, AES, WPA, WPA2
<b>Panda</b>	PAU09	Sí	2.4, 5	ac/b/g/n	Infraestructura	64/128-bit WEP, WPA/WPA2, WPA-PSK
<b>Rosewill</b>	N150	Sí	2.4	b/g/n	Ad-hoc	64/128-bit WEP, WPA, WPA2, WPA2-PSK, WPA-PSK, TKIP/AES
<b>Trendnet</b>	TEW-648UB N150 Mini Wireless USB Adapter	Sí	2.4	b/g/n		64/128-bit WEP, WPA/WPA2, WPA2-PSK, WPA-PSK,

Tabla 4. Modelos de adaptadores Wi-Fi para Raspberry Pi

## 2.4 Trabajos previos

Como se ha explicado en el apartado 1, el *leitmotiv* de este proyecto es la calibración del rendimiento del estándar IEEE 802.11 en espacios tridimensionales con redes ad-hoc, también llamadas IBSS. En este sentido, la proliferación de vehículos aéreos no tripulados y el amplio abanico de funciones que pueden realizar han generado gran cantidad de literatura que aborda la cuestión de las comunicaciones tierra-aire a través de IEEE 802.11, con nodos desplazándose en un espacio tridimensional. Así, el presente trabajo condensa el acervo de conocimientos en la materia recogidos por otros estudios y los aplica en el caso de un vehículo aéreo controlado por una Raspberry Pi.

La abundante literatura recopilada en torno a este asunto se justifica por las nuevas posibilidades que los estándares inalámbricos ofrecen en relación a la transmisión de datos en entornos donde el cable no puede llegar, pero también por los desafíos que supone en lo que refiere a conectividad y cobertura respecto a los medios cableados, inconvenientes que además se ven potenciados en el caso de transmisión de datos en vuelo [10]. En el común de las situaciones, con redes en las que su ancho y su largo son bastante superiores a su altura, una aproximación en dos dimensiones es correcta (ver Figura 3). Sin embargo, esta no va a ser la situación que se va a abordar. Para una red de  $N$  nodos, donde el dron está a una altura sobre el eje  $Y$  mayor que las dimensiones del plano  $XZ$  es necesario abordar una perspectiva tridimensional para así comprender la situación en toda su complejidad. Como se mencionó en el apartado 1, en estas circunstancias los nodos pueden encontrarse a distintas alturas y orientaciones, y ello no debe ser óbice para que las comunicaciones deban seguir siendo eficientes (ver Figura 4).

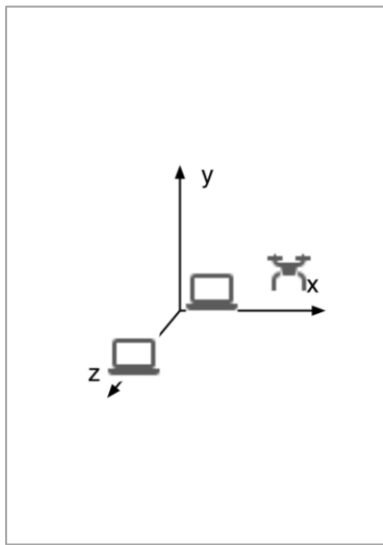


Figura 3. Modelo de red en 2D

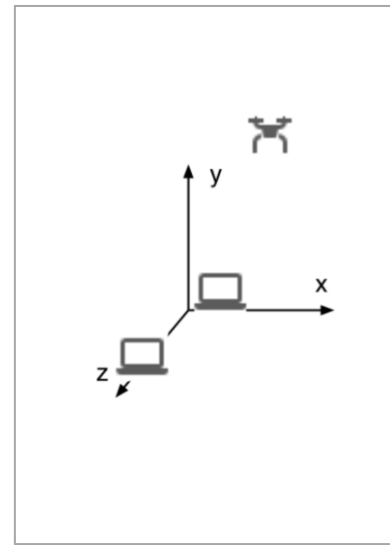


Figura 4. Modelo de red en 3D

Las señales inalámbricas son mucho más sensibles a factores externos que las guiadas por un medio de transmisión como un cable. Los entornos abiertos, donde controlar las condiciones ambientales es harto más difícil que en habitaciones confinadas, plantean multitud de variables (humedad, temperatura, fenómenos meteorológicos como lluvia o viento, presencia de obstáculos, etc.) a los que el estándar de comunicaciones debe sobreponerse para así cumplir con unos criterios de calidad en cuanto a caudal de transmisión, retrasos y pérdidas de paquetes. Verbigracia, en [12] se explica cómo detalles tan nimios (la especie de los árboles que se encuentran en un determinado entorno) influyen sobre la señal. Resulta evidente que la aportación individual de cada uno de estos elementos es, en correspondencia, insignificante, pero habida cuenta de la inmensa cantidad de factores despreciables que están presentes, el efecto total puede ser considerable. La altura a la que se encuentran dispuestas las antenas de los dispositivos modifica también el comportamiento de las señales. Esto es lo que se conoce como efecto paraguas. La imagen resulta ilustrativa: cuanto mayor sea la altura de la antena, hay una mayor cobertura debajo de la misma.

Como consecuencia de las variaciones que presenta el medio es arriesgado contar con un sistema electrónico de comunicación inalámbrica que conste de una única antena. Es por esta razón que los dispositivos que proporcionan conectividad inalámbrica hacen uso de la diversificación de antenas. Se trata de una técnica cuyos funcionamiento y resultados, aplicados a la comunicación entre una estación de tierra y una aeronave no tripulada, se detallan en [5]. En este paper se demuestra que la utilización de múltiples antenas transmisoras y/o receptoras incrementa de forma notable la robustez de los enlaces inalámbricos. Esta conclusión se basa en el hecho de que múltiples antenas colaborando dan lugar a una compensación del comportamiento individual de cada una de ellas: así, se experimentó con distintas rutas de vuelo y en el caso más favorable se produjo una mejora del 32% en los parámetros de calidad de la comunicación.

Entre otros trabajos que abordan las contingencias asociadas a las comunicaciones entre estación de tierra y un vehículo aéreo no tripulado, [13] propone un sistema de comunicación de relevos (*relay*) para situaciones donde cada UAV se encuentra más allá de la línea de visión del resto (*beyond-line-of-sight*, o BLOS por sus siglas en inglés). BLOS supone no sólo la imposibilidad de los sistemas para establecer comunicaciones entre ellos debido a que la distancia que los separa es superior al rango de alcance de la señal, sino también la presencia de obstáculos que mitigan la señal y hacen imposible que el receptor pueda decodificarla.



Figura 5. Sistema de comunicación inalámbrica basado en relevos

La Figura 5 ilustra un caso en el que, debido a la presencia de una alta montaña, el ordenador que constituye la estación de tierra no puede comunicarse directamente con el dron que se encuentra en la otra cara de la montaña. Es por ello que ambos necesitan de un UAV de apoyo situado sobre la cima de la montaña, con el que los sistemas transmisor y receptor gozan de visión directa y alcance.

Un proyecto relacionado íntimamente con el presente es [9]. En él, el dispositivo que hay a bordo del UAV es el modelo Galileo, resultado de la colaboración de Intel y Arduino. Se trata de un producto idóneo para tareas de sensado y monitorización con requerimientos de memoria que presenta sensibles diferencias con respecto a RPi: Galileo cuenta con un reloj propio y un procesador de 400MHz (las prestaciones del ordenador que será utilizado en este trabajo serán objeto de desarrollo en apartados posteriores).

Retomando los experimentos realizados con la placa Galileo, se trabajó con un UAV en situación de campo abierto, actuando como punto intermedio entre dos estaciones de tierra. Todas las comunicaciones se realizaron con estándares IEEE 802.11, adaptados automáticamente en función de las condiciones del entorno, y se tomaron mediciones de potencia de señal y velocidad de descarga para distintos escenarios (altura sobre el nivel del suelo del dron, modo de conexión utilizado, etc.). Entre las conclusiones que resultan de especial utilidad para este trabajo destaca el hecho de que un modo de conexión ad-hoc proporciona un rendimiento mayor cuando los paquetes a transmitir son grandes (por ejemplo, se pasó de un caudal de 4.5 Mb/s para paquetes de 512 bytes a 7.5 Mb/s para paquetes de 1024 bytes). Además, se constató que la corriente consumida por las baterías de Galileo es menor cuando el modo de conexión es ad-hoc. Una red ad-hoc se caracteriza por su simetría, es decir, cada nodo que la compone posee las mismas características funcionales (a diferencia de la configuración en infraestructura, donde se encuentran puntos de acceso y estaciones por separado, cada uno con una función diferenciada). En las redes ad-hoc los nodos suelen ser dispositivos portátiles muy dependientes de alimentación. En caso de que en esta configuración un nodo quiera transmitir a otro necesita orientar la antena y ajustar la potencia de la señal en función de las exigencias del entorno. A grandes rasgos, y considerando únicamente efectos de separación física entre puntos transmisor y receptor, la señal se debilita siguiendo una relación de  $1/d^a$ , donde  $d$  es la distancia entre nodos y  $a$  un factor de declive de la señal (o *signal decline factor*) que normalmente varía entre 2 y 4. Este fenómeno, a priori limitante y perjudicial, puede sin embargo convertirse en una virtud: en una red ad-hoc basta con que la potencia de la señal transmitida por un nodo sea suficiente como para llegar sin excesiva degradación al vecino más próximo, de forma que éste pueda decodificarla. Es decir, no es necesario alcanzar un AP a una distancia cualquiera, lo que en ciertas circunstancias puede suponer una atenuación intolerable de la señal. En definitiva, en las redes ad-hoc la potencia de transmisión de un nodo la determina principalmente la distancia al nodo vecino [16], consiguiendo una mayor eficiencia energética y, por ende, una mayor autonomía de las baterías, factor cuya relevancia en este proyecto es crucial.



# 3 HARDWARE Y SOFTWARE UTILIZADOS

*I do not fear computers. I fear the lack of them*

*- Isaac Asimov, bioquímico y escritor -*

En este apartado se detallarán los componentes del sistema de comunicaciones. Estos elementos necesitan de una configuración previa para ponerlos a punto de manera que se adecúen a las pretensiones que se persiguen, por lo que los ajustes necesarios a realizar sobre ellos también serán materia de desarrollo en los párrafos siguientes. En resumen, los dispositivos serán descritos a niveles tanto hardware como software, prestando especial atención a aquellos puntos que más relevantes resulten en relación con los objetivos del proyecto y a aquellas fases de la configuración que hayan supuesto escollos.

## 3.1. Raspberry Pi

La tecnología de los vehículos aéreos no tripulados no es la única que ha experimentado un proceso de *democratización* en su uso. Sin ir más lejos, el dron incorporará un ordenador que, por su bajo precio, su tamaño reducido y su peso liviano resulta idóneo como controlador de este tipo de producto.

En un principio, las Raspberry Pi se comercializaron con el objetivo de proporcionar a los niños un ordenador de bajo coste con el que poder dar sus primeros pasos en programación. Sin embargo, sus características prestaciones pronto la convirtieron en una buena opción para aplicaciones que no precisaban del consumo de un ordenador de sobremesa o similar para su funcionamiento. En los albores del proyecto se optó por usar el modelo de Raspberry Pi 2 Model B, si bien las mayores prestaciones del modelo inmediatamente superior, Raspberry Pi 3 Model B, hicieron decantarse por este último. En este sentido, Raspberry Pi 3 Model B aporta una novedad respecto a su predecesora que resulta muy interesante en el desarrollo de este proyecto. Este salto cualitativo consiste en sendos módulos Wi-Fi y Bluetooth integrado. La evaluación de la calidad de conexiones por Bluetooth escapa del objetivo de este proyecto; por el contrario, el hecho de disponer de una interfaz Wi-Fi integrada permitirá comparar su potencial respecto al del *dongle* USB.

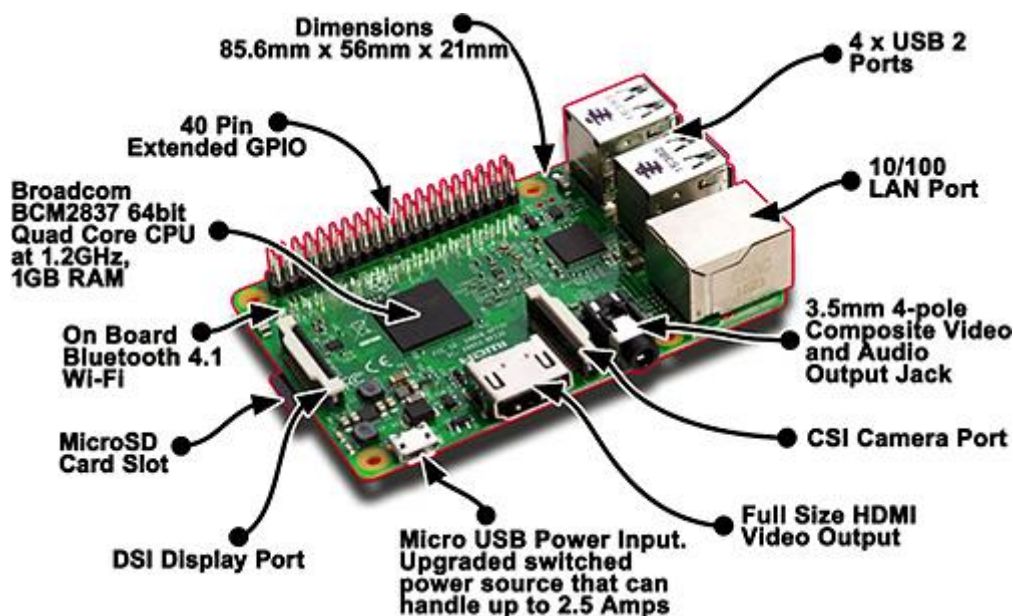


Figura 6. Componentes de la Raspberry Pi Model 3

Raspberry Pi 3 Model B cuenta con un procesador de cuádruple núcleo a 1200 MHz de frecuencia de trabajo, 1 gigabyte de memoria RAM, cuatro puertos USB (de los cuales uno se destinará al dongle elegido en el apartado 2.3), cuarenta pines que permiten la incorporación de periféricos que añaden funcionalidad al sistema, tales como convertidores digital-analógico o analógico-digital u otros que resultan especialmente indicados para aplicaciones de vuelo: sensores atmosféricos, barométricos, infrarrojos, módulos GPS, etc. La instalación de estos accesorios no reviste gran dificultad y se encuentra suficientemente documentada en [19].

	<b>RPi 3 Model B</b>	<b>RPi 2 Model B</b>
<b>System on Chip</b>	Broadcom BCM2837	Broadcom BCM2836
<b>CPU</b>	ARM Cortex-A53 4 núcleos 1200 MHz	Cortex-A7 4 núcleos 900 MHz
<b>GPU</b>	VideoCore IV multimedia @ 400 MHz	Broadcom VideoCore IV @ 250 MHz
<b>RAM</b>	1 GB	1 GB
<b>GPIO</b>	40	40
<b>USB 2.0</b>	4	4
<b>Entrada de vídeo</b>	Conector para interfaz CSI de 15 pines MIPI	Conector para interfaz CSI de 15 pines MIPI
<b>Salida de vídeo</b>	Jack de 3,5 mm para HDMI y vídeo compuesto (PAL y NTSC)	Jack de 3,5 mm para HDMI y vídeo compuesto (PAL y NTSC)
<b>Almacenamiento</b>	microSD	microSD
<b>Ethernet</b>	10/100 Mbps	10/100 Mbps
<b>Tamaño</b>	85,60 x 56,5 mm	85,60 x 56,5 mm
<b>Peso</b>	45 gramos	45 gramos

Tabla 5. Comparativa de modelos 2B y 3B

Raspberry 3B supera a su inmediato predecesor en lo que refiere a la velocidad. Este hecho, que a la vista de los datos referentes a la CPU en la Tabla 5 podía suponerse, se demuestra empíricamente en un estudio basado en casos de prueba estandarizados para la medición del rendimiento en este tipo de dispositivos. En concreto, [17] recurre a los tests de Dhrystone (capacidad de realizar operaciones en base entera, medida en millones de instrucciones por segundo o MIPS) y Linpack (análogo a Dhrystone pero basado en coma flotante, y medido en millones de operaciones en coma flotante por segundo o MFLOPS).

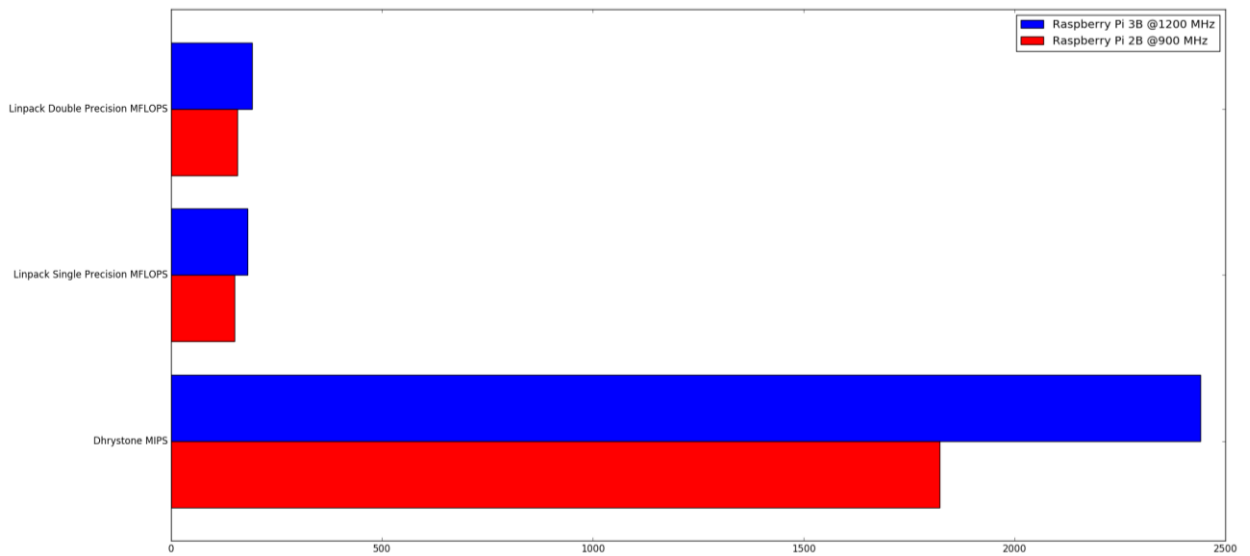


Figura 7. Comparativa del rendimiento de 2B y 3B de acuerdo con los tests de Linpack y Dhrystone

Esta diferencia de potencial entre los dos productos se apreciará mejor con el cálculo de la criba de Eratóstenes (algoritmo que permite hallar todos los números primos menores a uno dado, en este caso un millón): mientras que 2B realiza dicha operación en 2:05 minutos, 3B obtiene todos los resultados en 1:07 minutos, lo que supone una reducción en el tiempo de cálculo de casi un cincuenta por ciento.

Obviamente, el incremento de la capacidad operativa de 3B tiene una desventaja, y es el aumento del consumo. No es casual que, en el ámbito de la electrónica, las sucesivas mejoras en los dispositivos traigan aparejadas la creciente necesidad de pequeñas baterías que sean capaces de suministrar a éstos la energía que necesitan para funcionar. En [17] también se hace referencia a esta diferencia de consumo entre distintas versiones de RPi.

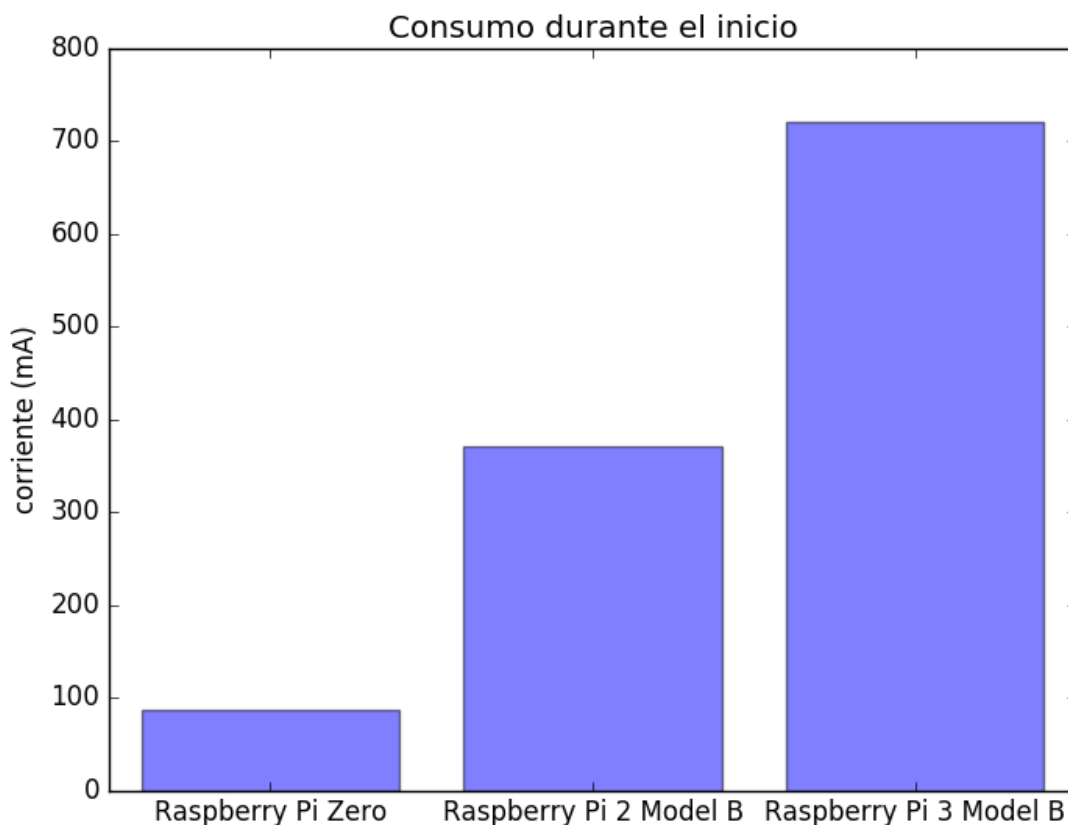


Figura 8. Máximo consumo de modelos de RPi durante la puesta en marcha del dispositivo

En la Figura 8 se aprecia un incremento de prácticamente un cincuenta por ciento entre la máxima corriente consumida por 2B y 3B durante el inicio.

Las prestaciones de Raspberry Pi pueden verse aumentadas mediante el uso de pines. Por ello, un aspecto al que hay que prestar especial atención a la hora de trabajar con la Raspberry Pi es la comprensión de las funciones y capacidades de los cuarenta pines de propósito general (GPIO) que incorpora. El desconocimiento de las particularidades de cada uno de ellos puede llevar a que el equipo no responda como se espera o, en el peor de los casos, que los pines resulten dañados de forma irremisible. Por el contrario, un buen uso de los pines permitirá integrar en la Raspberry Pi numerosos dispositivos que aumentarán el potencial del sistema: higrómetros, termómetros, relojes, etc.



Figura 9. Diagrama de pines en Raspberry Pi 3 Model B

Pin	Nombre	Comentarios	Pin	Nombre	Comentarios
1	3V3	Alimentación 3.3V	21	GPIO9	SPI0_MISO
2	5V	Alimentación 5V	22	GPIO25	
3	GPIO2	SDA1 I2C	23	GPIO11	SPI0_SCLK
4	5V	Alimentación 5V	24	GPIO8	SPI0_CE0_N
5	GPIO3	SCL1 I2C	25	Ground	Tierra
6	Ground	Tierra	26	GPIO7	SPI0_CE1_N
7	GPIO4		27	ID_SD	I2C ID EEPROM
8	GPIO14	UART0_TXD	28	ID_SC	I2C ID EEPROM
9	Ground	Tierra	29	GPIO5	
10	GPIO15	UART0_RXD	30	Ground	Tierra
11	GPIO17		31	GPIO6	
12	GPIO18	PCM_CLK	32	GPIO12	
13	GPIO27		33	GPIO13	
14	Ground		34	Ground	Tierra
15	GPIO22		35	GPIO19	
16	GPIO23		36	GPIO16	
17	3V3	Alimentación 3.3V	37	GPIO26	
18	GPIO24		38	GPIO20	
19	GPIO10	SPI0_MOSI	39	Ground	Tierra

20	Ground		40	GPIO21	
----	--------	--	----	--------	--

Tabla 6. Nombre y funcionalidad de los pines de la Raspberry Pi 3 Model B

Como se refleja en las Figura 9 y Tabla 6, entre los pines de la Raspberry Pi 3 Model B, cuatro se destinan a su alimentación: dos trabajan a 3,3 voltios y los otros dos restantes a 5 voltios. Estos últimos se utilizan, por ejemplo, con aplicaciones basadas en Arduino.

### 3.1.1. Instalación de la imagen Raspbian, actualización de software y ajustes menores

El sistema operativo con el que trabajará la Raspberry Pi se denomina Raspbian. Se trata de un software de código abierto basado en Debian Linux. Raspbian es mantenido por un grupo de desarrolladores que son usuarios de Raspberry Pi, pero no es un producto perteneciente a la Raspberry Pi Foundation. La versión de Raspbian a utilizar es Jessie, que ofrece un atractivo y accesible entorno gráfico, o GUI por sus siglas en inglés, basada en escritorio.

El nombre del usuario es *pi* para ambas Raspberry Pi. La contraseña de ambas también es común: *raspberr*. Una vez instalada la imagen del sistema operativo, previamente descargada y guardada en la tarjeta Micro SD, conviene realizar una serie de operaciones de puesta a punto del sistema. Otra práctica conveniente es modificar el *layout* del teclado para adaptarlo a la lengua castellana.

### 3.1.2. Manejo de Raspberry Pi desde PC y conexión a la red

La configuración referida en líneas anteriores puede considerarse accesoria. Sin embargo, más importante es conseguir conectividad de la Raspberry Pi con la red mediante Ethernet. Esto se consigue asignando una dirección IP estática al dispositivo. Para ello es necesario modificar, con permisos de superusuario, el fichero */etc/network/interfaces*, tal y como se indica en la Figura 10.

```

auto lo
iface lo inet loopback

iface eth0 inet static
address 192.168.137.176
netmask 255.255.255.0
network 192.168.137.0
broadcast 192.168.137.255
gateway 192.168.137.1

auto lo
iface lo inet loopback

iface eth0 inet static
address 192.168.137.175
netmask 255.255.255.0
network 192.168.137.0
broadcast 192.168.137.255
gateway 192.168.137.1

```

Figura 10. Configuración de interfaces de las respectivas RPi

Para que los cambios hagan efecto basta con introducir desde consola el comando *reboot*. Una vez reiniciado el dispositivo puede comprobarse que las direcciones estáticas de cada una de las interfaces son aquellas que se desean. Análogamente a la configuración estática de las direcciones IP de las interfaces cableadas, también las direcciones de las interfaces Wi-Fi pueden fijarse. Así, a la Raspberry Pi con dirección de red estática Ethernet 192.168.137.175 se le asigna la dirección estática Wi-Fi 192.168.1.137, y la RPi con dirección Ethernet 192.168.137.176 tendrá asociada la dirección Wi-Fi 192.168.1.138 (ver Tabla 7). Como se verá luego, esto resultará de gran interés pues permitirá probar el funcionamiento de la aplicación de comunicaciones de forma rápida y cómoda.

Dispositivo	Dirección red Ethernet	Dirección red Wi-Fi
RPi 1	192.168.137.175	192.168.1.137
RPi 2	192.168.137.176	192.168.1.138

Tabla 7. Configuración de red de las Raspberry Pi

Una vez asignadas las direcciones de red estáticas en sendas interfaces Ethernet se pueden manejar las RPi a través del propio ordenador mediante el software de acceso remoto VNC. Instalando el servidor VNC en la RPi, es posible acceder remotamente a ellas si se dispone del cliente VNC en el ordenador de trabajo.

La configuración por defecto de Raspbian hace necesario abrir expresamente el servidor VNC a través de línea de comandos cada vez que se enciende la RPi. Para facilitar el trabajo y acceder directamente desde el ordenador a la Raspberry Pi cuando se quiera utilizar, es posible forzar el inicio automático del servidor mediante el script mostrado en la **Error! Reference source not found.**, localizado en el directorio `~/config/autostart`. Como puede deducirse de su nombre, en dicho directorio se encuentran todos los scripts escritos por el administrador y que se ejecutan de forma sistemática al encender la RPi.

```

[Desktop Entry]
Type=Application
Name=TightVNC
Exec=vncserver :1 -geometry 1280x800 -depth 16 -pixelformat rgb565
StartupNotify=false

```

Figura 11. Script que fuerza el arranque del servidor VNC al encender Raspberry Pi

Si además se quiere aprovechar la conexión a internet del PC para dar acceso a la red a Raspberry Pi (que es, al fin y al cabo, lo que se busca con toda la configuración previa que se ha realizado) se debe dar uso compartido a la interfaz Wi-Fi del PC.

En primer lugar, se modifican las propiedades de la interfaz inalámbrica, marcando la opción que permite dar uso compartido a esta interfaz con otra, en este caso Ethernet (ver Figura 12). A continuación, en las propiedades de la interfaz cableada, se le asigna una dirección IPv4 que se encuentre en el rango de la que previamente se asignó a Raspberry Pi. A la vista de la Figura 10 se concluye que 192.168.137.1 es una dirección que cumple estas características (ver Figura 13).

Una vez ejecutados estos cambios Raspberry Pi está conectada a internet. En sistemas Windows se recomienda que cada vez que se quiera compartir la conexión del ordenador con Raspberry Pi se desmarque la opción de conexión compartida (en caso de que esté ya por defecto siempre señalada) y se vuelva a marcar, pues se han dado casos en los que no basta con conectar el cable de red RJ45 entre ambos puntos para proporcionar conectividad.

Conseguida la conectividad a la red la operación primordial es mantener a punto la imagen del sistema. Para ello es necesario trabajar con privilegios de administrador (con el comando `sudo`) y ejecutar desde consola, secuencialmente, `apt-get update`, que actualiza la lista de paquetes del sistema; y posteriormente `apt-get upgrade` para que todos los paquetes previamente instalados permanezcan en su última versión. Esta tarea puede prolongarse durante varios minutos, especialmente cuando se realiza de forma inmediatamente posterior a la instalación de la imagen del sistema. Efectuando esta tarea con periodicidad se garantiza que la imagen del sistema se mantiene siempre actualizada.

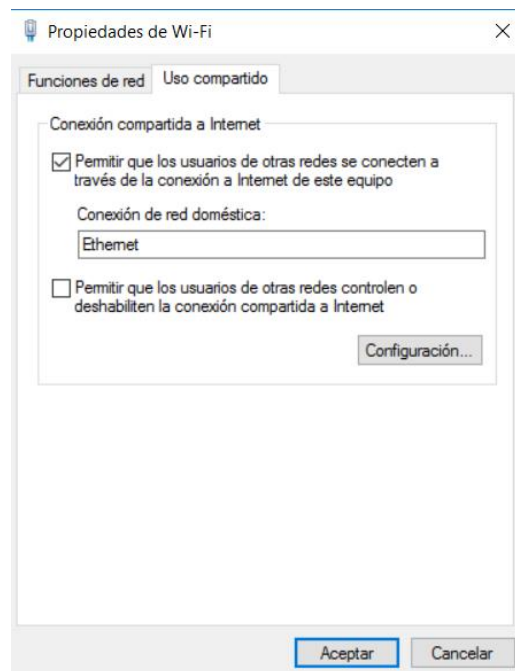


Figura 12. Opción que permite compartir la conexión inalámbrica con la interfaz Ethernet

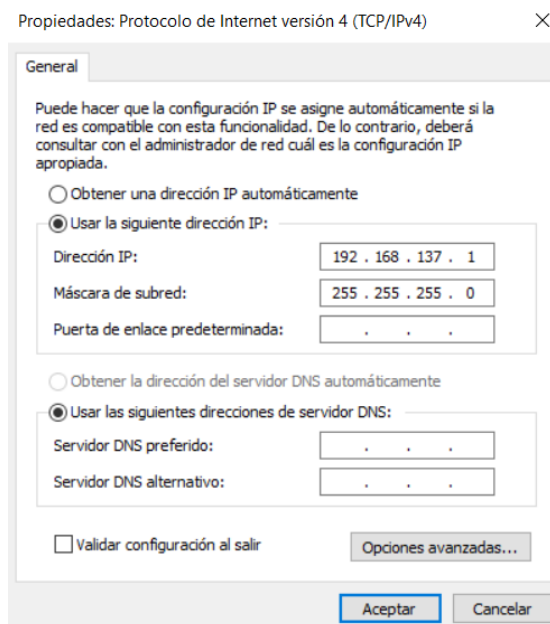


Figura 13. Direccionamiento proporcionado a la interfaz Ethernet

### 3.1.3. Despliegue de la red ad-hoc

Una red ad-hoc es un tipo de red descentralizada en la que los nodos se comunican entre sí sin necesitar de una estación base o un punto de acceso cableado. En comparación con las redes basadas en infraestructura, los sistemas ad-hoc se caracterizan por:

- movilidad superior. De hecho, en las redes ad-hoc no hay necesariamente elementos con una posición fija, situación que sí se da en el caso de los puntos de acceso cableados. Es por ello que a menudo se utiliza el acrónimo MANET (*mobile ad-hoc networks*) para referirse a estas redes.
- topología variante, debida precisamente a la movilidad de las redes ad-hoc.
- estructura descentralizada, distribuida, en la que aspectos como la seguridad y el *routing* no están

controlados por un dispositivo *master* concreto.

- robustez, debida precisamente a la estructura distribuida de la red, que en caso de que un nodo fallara no la dejaría inutilizada, como sí sería en caso de contar con un punto de acceso que no cumpliera con sus funciones debidamente.
- menor complejidad en la instalación: sólo es necesario realizar unos ajustes en software, explicados posteriormente, para conseguir que varios dispositivos compartan una red.
- menor coste y mayor rentabilidad en entornos habitualmente despoblados, donde los puntos de acceso estarán durante la mayor parte del tiempo inutilizados [26],

Todas estas particularidades hacen de las redes ad-hoc una opción interesante en aplicaciones militares, industriales o de rescate (imagínese que es necesario llevar a cabo una operación de este tipo en un medio que no cuenta con una infraestructura Wi-Fi previa, por ejemplo, una zona escarpada de una alta montaña). En lo que atañe a este proyecto, las RPi colocadas sobre la estación de tierra y el vehículo aéreo formarán un IBSS, siglas con las que también se designa a las redes ad-hoc.

Con las direcciones de las interfaces inalámbricas configuradas en el apartado anterior es posible conformar una red ad-hoc que proporcione conectividad a las dos RPi. En primer lugar y como es obvio, los *dongles* deberán soportar este modo de conexión para llevar a buen término la operación. Cumplido este requisito la red puede configurarse mediante unos ajustes sencillos en el fichero `/etc/network/interfaces`, que se muestran en la Figura 14. En concreto, estos cambios atañen a los valores de *wireless-mode*, que debe ser *ad-hoc*; y, por otro lado, a *wireless-channel* y *wireless-essid*, que en ambos nodos deben ser idénticos para que haya conexión entre ellos.

```
auto wlan0
iface wlan0 inet static
    address 192.168.1.138
    netmask 255.255.255.0
    wireless-channel 1
    wireless-essid MYNETWORK
    wireless-mode ad-hoc
```

Figura 14. Configuración de la interfaz WLAN

La línea `auto wlan0` provoca que esta interfaz se active inmediatamente al arrancar el sistema, haciendo que la conexión a la red ad-hoc se realice automáticamente. Esto se puede corroborar posicionando el cursor sobre el icono de gestión de conexiones en la barra superior de Raspbian. Si todo el proceso se ha desarrollado correctamente, debería mostrarse lo indicado en la Figura 15.

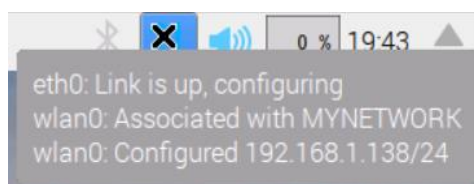


Figura 15. Visualización de conexiones activas en Raspbian

Llegados a este punto, ambas RPi son capaces de comunicarse entre ellas sin necesidad de que ninguna de ellas esté conectada a internet, como se puede corroborar a partir de la Figura 16, que muestra el resultado de un ping realizado desde 192.168.1.137 con destino 192.168.1.138. Esto supone un avance clave en el proyecto, pues propicia la consecución de las comunicaciones entre cliente y servidor cuya calidad se evaluará posteriormente. No obstante, es imprescindible para la consecución de los objetivos de este proyecto que una de las RPi, en concreto la que actúa en la estación de tierra, conste de al menos una conexión a internet, pues



así se logrará que ambas estén sincronizadas.

```
pi@raspberrypi:~ $ ping 192.168.1.138
PING 192.168.1.138 (192.168.1.138) 56(84) bytes of data:
64 bytes from 192.168.1.138: icmp_seq=1 ttl=64 time=0.949 ms
64 bytes from 192.168.1.138: icmp_seq=2 ttl=64 time=0.825 ms
64 bytes from 192.168.1.138: icmp_seq=3 ttl=64 time=0.781 ms
64 bytes from 192.168.1.138: icmp_seq=4 ttl=64 time=0.790 ms
64 bytes from 192.168.1.138: icmp_seq=5 ttl=64 time=0.806 ms
```

Figura 16. Pruebas de conectividad en la red ad-hoc

Como se ha explicado anteriormente, Raspberry Pi 3 cuenta con un módulo Wi-Fi incorporado. La conectividad inalámbrica mediante este sistema puede configurarse a voluntad comentando o descomentando el siguiente comando: `dtoverlay=pi3-disable-wifi`, en el fichero `/boot/config.txt`. Es recomendable que, cuando se vayan a hacer las pruebas con el dongle USB, se descomente dicho comando, pues en caso contrario el módulo Wi-Fi integrado (o *built-in*) tendrá preminencia sobre el dongle.

### 3.1.4. Sincronización de fecha y hora: el protocolo NTP

Una de las razones que hacen de Raspberry Pi un producto al alcance de todos los bolsillos es su carencia de algunas prestaciones básicas. Una de ellas es el reloj interno (RTC): este dispositivo, si bien es bastante económico (alrededor de 10 USD) es, en términos relativos respecto al coste de Raspberry Pi, caro, pues supondría cerca de un 20 por ciento de su precio de venta.

Si el RTC no está presente, la Raspberry Pi precisa de una conexión a la red para conocer cuál es la hora y la fecha. Esto lo realiza a través del protocolo NTP (Network Time Protocol) [28], gracias al cual el cliente se conecta a un servidor que le proporciona periódicamente esta información. NTP es un protocolo antiguo, datado de 1985, y hoy día su uso es generalizado. Utiliza UDP como capa de transporte y el puerto 123. El motivo de usar un protocolo no orientado a conexión es que en internet el tiempo es un factor crítico, y por tanto se precisa de una conexión rápida, sin apenas latencia. NTP permite que todos los equipos pertenecientes a una misma red reciban la misma señal de reloj. Cabría preguntarse, entonces, cómo especificar a qué servidor hay que conectarse para obtener la hora correcta de acuerdo a la localización geográfica. En [www.pool.ntp.org/](http://www.pool.ntp.org/) se recopilan todos estos servidores y se clasifican según la zona del planeta a la que dan servicio. En el momento de escribir estas líneas, en Europa hay unos dos mil ochocientos servidores NTP. La precisión de los mismos depende de quién le proporciona la información sobre la hora y fecha (ver Figura 17).

Es en este momento cuando resulta pertinente explicar el concepto de estrato – *stratum*-. Se define como una medida de la distancia de sincronización, definida como el número de servidores que hay hasta llegar a un reloj de referencia. Así, los relojes atómicos de muy alta precisión constituyen el estrato 0, son la referencia primaria de información, y, al mismo tiempo, la más escasa por el alto coste que supone adquirir y mantener sistemas basados en relojes atómicos de cesio o rubidio, receptores GPS y CDMA. Se caracterizan por no presentar retraso y, si lo hubiere, de una magnitud infinitesimal. Los servidores que reciben información UTC del estrato 0 a través de radio, satélite o módem y la proporcionan a otros clientes de nivel inferior constituyen el estrato 1. Constituyen, además, el primer nivel de servidores NTP insertos en la red. Estos clientes que recibieron información del nivel 1 componen el estrato 2, y a su vez son servidores para clientes del estrato 3 [25], y así sucesivamente hasta el estrato 16. Un servidor sincronizado con otro de estrato  $n$  formará parte del estrato  $n+1$ . Cada estrato debe incluir varios dispositivos, es decir, se busca la redundancia en caso de fallo. Un reloj que se encuentra en el estrato 16 se considera inalcanzable, y en caso de que así sea será necesario buscar otros más cercanos. En este sentido recuerda a protocolos de routing como RIP, que utilizan el valor de métrica 16 para señalar que un punto de la red es inaccesible.

De una red NTP se dice, utilizando un término anglosajón, que es *semi-self-organizing*. Esto significa que todos los nodos pertenecientes a la red precisan de una configuración manual previa que, una vez realizada, hace que funcionen según se requiere de forma automática.

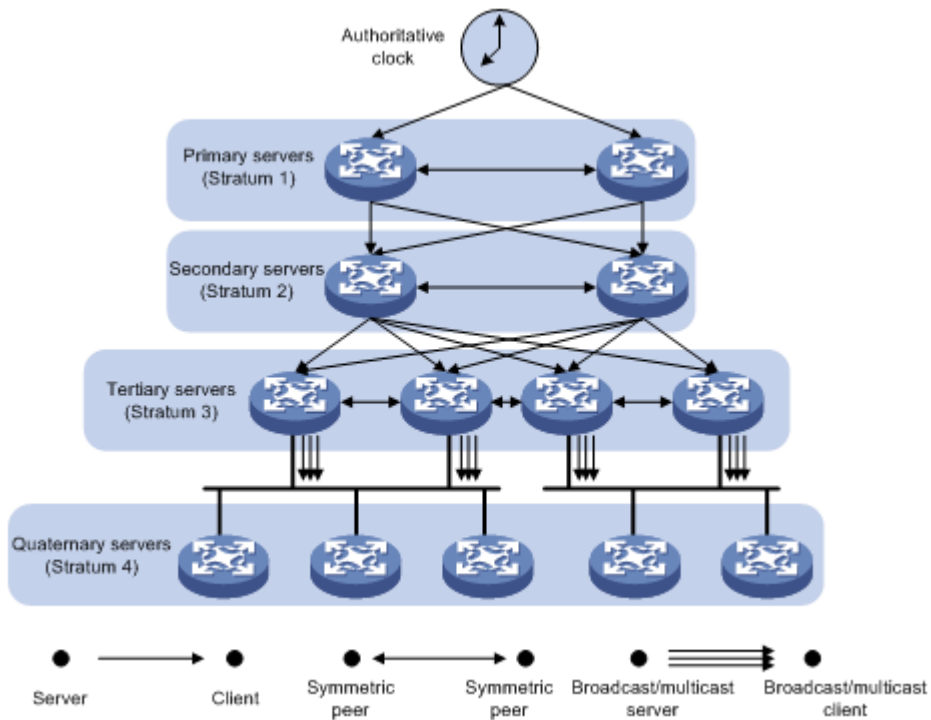


Figura 17. Jerarquía de servidores NTP

```
# pool.ntp.org maps to about 1000 low-stratum NTP servers. Your server will
# pick a different set every time it starts up. Please consider joining the
# pool: <http://www.pool.ntp.org/join.html>
server hora.roa.es iburst
server 0.europe.pool.ntp.org
server 1.europe.pool.ntp.org
server 2.europe.pool.ntp.org
server 3.europe.pool.ntp.org
```

Figura 18. Direcciones de los servidores en el fichero /etc/ntp.conf

Dentro del fichero /etc/ntp.conf se puede listar, con permisos de súperusuario, las direcciones de los servidores de los que se puede obtener la fecha y hora (ver Figura 18). Es práctica recomendable incluir al menos dos direcciones para que los errores en un servidor no resulten críticos y dejen al sistema sin referencia horaria. Sirva como ejemplo el servidor con dirección hora.roa.es, situado en la localidad gaditana de San Fernando, que resultaría adecuado por criterios de proximidad geográfica.

Es necesario, además, realizar otro cambio para conseguir que la Raspberry Pi que no está conectada a internet mediante la interfaz Ethernet reciba de la RPi conectada la información horaria. Dicha variación se muestra en la Figura 19. Se trata de descomentar, en el fichero /etc/ntp.conf de la RPi no conectada a internet, sus dos últimas líneas. Con esto lo que se consigue es que la RPi escuche mensajes de difusión NTP dentro de la WLAN y actualice su información horaria en función de estos mensajes.

```
# If you want to listen to time broadcasts on your local subnet, de-comment the
# next lines. Please do this only if you trust everybody on the network!
disable auth
broadcastclient
```

Figura 19. Modificación en /etc/ntp.conf para escuchar mensajes de difusión NTP

El comando `ntpq -pn` muestra revela parámetros sobre los datos que se están recibiendo (ver Figura 20). En la primera columna se observa que, a la izquierda de las direcciones de los servidores, hay una serie de símbolos. Los asteriscos indican que es éste el servidor del que se está tomando la información en este momento. El símbolo “+” refiere a que ese servidor actúa como respaldo en caso de que el principal caiga. Por último están los marcados con “-”, que no cumplen en este momento ninguna función, pero pueden pasar a ser servidores de respaldo en cualquier momento. La columna “t” indica la categoría (el estrato) de cada uno de los servidores. Se observa que el servidor NTP en uso es de estrato 1, es decir, que está directamente comunicado con una fuente primaria. “Delay” se refiere al *round-trip time* (RTT), o tiempo necesario desde que un pulso viaja desde una fuente hasta su destino para luego hacer el camino inverso. “Offset” indica la diferencia en milisegundos entre la hora ofrecida por el servidor y la del sistema. “Jitter” es la diferencia hallada entre dos muestras consecutivas. Se trata, por tanto, de un sistema que no es perfecto, pues depende fuertemente de la calidad de la red y de la precisión de los relojes que aportan la información. Con todo, es sin duda la opción idónea en caso de que los nodos de la comunicación no cuenten con un reloj interno, como es el caso.

Hay que dejar pasar un cierto intervalo, en torno a dos minutos, para conseguir que todos los nodos de la red se sincronicen. NTP es, en ocasiones, un protocolo con un tiempo de convergencia prolongado hasta que los equipos a la escucha de tramas NTP actualicen sus tablas y actualicen sus datos horarios. Esto se debe a que las tramas de difusión UDP proporcionan información muy precisa pero no demasiado frecuentemente, pues se envían aproximadamente cada minuto. Si pasado un tiempo prudencial siguiera sin haber sincronización se recomienda ejecutar, con permisos de súperusuario, el comando `“/etc/init.d/ntp restart”`. Con respecto a este tema, se comprobó que la sincronización horaria en el modelo Raspberry Pi 3 es mucho más rápida que en el 2B.

Puede concluirse que cuanto menor sea la cifra que refleje el campo “st” en el servidor, más fidedigna será la información que se recibe de él. Cifras altas, cercanas a 16, son muy desaconsejables, pues no aseguran precisión.

```
pi@raspberrypi:~ $ ntpq -pn
remote          refid           st t when poll reach  delay  offset  jitter
-----
+81.19.96.148   145.238.203.14  2 u  11  128  377  42.434 -6.127  2.851
*158.227.98.15  .GPS.           1 u  75  128  377  56.430 -9.784  4.315
```

Figura 20. Resultado de ejecutar `ntpq -pn`

Se ve, por tanto, que NTP depende irremisiblemente de la comunicación dentro de una red. En caso de que la conexión a internet cayera, Raspberry Pi puede seguir ofreciendo la hora en función de la información que recibiera del servidor NTP antes de la desconexión. Sin embargo, una vez que Raspberry Pi se apaga toda esta información se pierde, y al volver a encenderla se comprueba que la hora se ha desconfigurado. Es en este punto donde la red ad-hoc configurada previamente entra en juego: puesto que una de las RPi, la que está dispuesta sobre el dron, no va a tener acceso a internet, su única fuente de información horaria es la RPi asociada a la estación de tierra. Si esta última tiene conexión a un servidor NTP actuará como servidor horario de la RPi del UAV.

En el caso del proyecto, la RPi conectada a internet, y por tanto la que actuará como servidor NTP para la red ad-hoc, será la de dirección de red 192.168.1.137. La RPi con 192.168.1.138 sincronizará su información horaria con esta última<sup>2</sup>.

### 3.2. Alimentación

Dados el entorno en el que se van a desarrollar las pruebas y las necesidades de movilidad de las Raspberry Pi, éstas precisan de unas fuentes de alimentación externas que aseguren su disponibilidad durante el tiempo en el que se desarrollen las pruebas. Se utilizaron sistemas de alimentación distintos según se tratasen de pruebas estáticas o dinámicas. Para las primeras se utilizó la batería Poweradd MP-1311RR Slim2, cuyas características figuran en la Tabla 8.

<sup>2</sup> Para repasar la configuración de red de cada una de las Raspberry Pi se recomienda acudir a la Tabla 7.

Parámetros de la alimentación	
Tipo de batería	VideoCore IV multimedia @ 400 MHz
Capacidad	5000 mAh
Ciclo de vida útil	500 cargas
Voltaje y corriente de entrada	DC 5V / 2A
Voltaje y corriente de salida	DC 5V / 2.1A

Tabla 8. Características de la batería Poweradd MP-1311RR Slim2

Para ejecutar las pruebas dinámicas, la Raspberry Pi fue alimentada utilizando la conexión que la batería del UAV utilizado proveía, de forma que esta última alimentaba tanto al sistema de comunicaciones Wi-Fi como a todo el circuito que conformaba el vehículo aéreo.

### 3.3. Dispositivos de conexión inalámbrica

Con respecto a todos los dispositivos referenciados en la Tabla 4, el producto Wi-Pi de Element14 reúne varias características interesantes: su especificidad (está diseñado expresamente para proporcionar conectividad IEEE 802.11 a una Raspberry Pi, asegurando compatibilidad con el sistema operativo utilizado y soporte especializado), su compatibilidad con el modo de conexión ad-hoc, así como su bajo coste en comparación con dispositivos similares hacen de él una solución apta. En lo que respecta a los modos de conexión utilizados y estándares soportados, estos son ad-hoc y 802.11 b/g/n respectivamente, no suponiendo estos aspectos una diferencia cualitativa respecto a los que presentan los demás productos examinados. En lo que respecta a sus condiciones de trabajo, tanto la temperatura (-40 a 70°C) como la humedad ambiental relativa (10-90%) establecidas para su funcionamiento con garantías se dan con creces en los entornos donde se testará la aplicación. En situaciones donde estos dispositivos podrían estar sometidos a estrés, es decir, aquellas en las que las características del dispositivo son testadas al límite, véase incendios, sería imprescindible idear un acondicionamiento para los mismos que permitiera un correcto funcionamiento.

Las características del dongle utilizado pueden examinarse mediante el comando `sudo lshw -class network`. En el caso del dongle Wi-Pi, arroja los resultados que se muestran en la Figura 21. Entre la información que refleja dicha figura, interesa especialmente conocer el estándar IEEE 802.11 que está utilizando. En este caso es IEEE 802.11n.

```
pi@raspberrypi:~$ sudo lshw -class network
*-usb:1
  description: Wireless interface
  product: 802.11 n WLAN
  vendor: Ralink
  physical id: 4
  bus info: usb@1:1.4
  logical name: wlan0
  version: 1.01
  serial: 1.0
  capabilities: usb-2.00 ethernet physical wireless
  configuration: broadcast=yes driver=rt2800usb driverversion=4.9.35-v7+ firmware=0.29 ip=192.168.1.138 link=yes maxpower=450mA multicast=yes speed=480Mbit/s wireless=IEEE 802.11
```

Figura 21. Características del dongle Wi-Pi

Como se ha comentado en el apartado 3.1, se comparará el rendimiento del dongle Wi-Pi con el transmisor integrado del que dispone Raspberry Pi 3B. Este también hace uso del estándar IEEE 802.11n, hecho que asegura uniformidad en las condiciones del experimento y por tanto unos resultados más fidedignos que en el caso de que se usaran estándares distintos. Además, como se verá en páginas posteriores, el uso de IEEE 802.11n tiene implicaciones en los efectos del fenómeno de interferencia electromagnética sobre la calidad de la señal inalámbrica.

## 3.4. Elementos software

### 3.3.1. Lenguaje de programación: Python

Python es un lenguaje de programación lanzado en 1991 por el científico de la computación neerlandés Guido von Rossum. Entre sus características, pueden destacarse:

- Su simplicidad, basada en una sencilla sintaxis, similar a la de C.
- Su condición de software libre y de código abierto (*open source*), sustentado por una comunidad de programadores.
- El uso de un lenguaje de alto nivel (HLL).
- Su portabilidad, de modo que puede utilizarse en una gran cantidad de sistemas operativos, entre ellos Linux, Windows y Mac.
- Es interpretado, es decir, no requiere de una compilación a lenguaje máquina.
- Está orientado a objetos.

### 3.3.2. Entorno de desarrollo: Anaconda

Anaconda es una distribución libre de Python respaldada por una comunidad de desarrolladores y que dispone de una colección de unos 720 paquetes de código abierto. Este software, sencillo de instalar, supuso el primer contacto con lenguaje Python. Anaconda dispone de Spyder, un entorno de desarrollo interactivo que permite hacer labores de edición, testeo y debugging de forma simple, y que además realiza tareas de computación numérica valiéndose de librerías de Python como *matplotlib*, que será indispensable para la representación gráfica de los datos obtenidos durante el examen de las comunicaciones inalámbricas.

### 3.3.3. Programa cliente-servidor basado en TCP y UDP

Para realizar las comunicaciones entre la estación de tierra y el UAV se escribieron programas cliente y servidor, ambos basados en dos protocolos de transporte distintos: TCP y UDP. Puesto que está suficientemente demostrado [22], [24] que según el protocolo de transporte utilizado se obtienen distintos resultados, la motivación de usar ambos radica en el deseo de ampliar y perfeccionar la caracterización de las comunicaciones inalámbricas. Los programas servidores de ambos protocolos (*servidortcp.py* y *servidorudp.py*) están configurados de tal forma que muestren un log con el contenido y la hora de recepción de los mensajes (ver Figura 22). Estos ficheros logs constituyen un respaldo a las capturas de Wireshark, ya que indican de forma explícita qué mensajes se han recibido desde el cliente (en las capturas de Wireshark, para ver el contenido de las tramas, hay que hacer una inspección mucho más exhaustiva).

```
cd /
cd /home/pi/Desktop
sudo python servidortcp.py | tee -a $(date +%Y_%m_%d_%L%M%p)logtcp.txt & python servidorudp.py | tee -a $(date +%Y_%m_%d_%L%M%p)logudp.txt
```

Figura 22. Script que ejecuta el arranque del servidor

TCP es un protocolo orientado a conexión, seguro, con control de flujo y congestión y monitorización de la comunicación. Se trata de un protocolo de tráfico elástico, es decir, se adapta a las limitaciones de la red. Que un protocolo sea orientado a conexión implica que precisa del establecimiento de una conexión lógica entre nodos (*three-way handshake*) antes de efectuarse el intercambio de información y, asimismo, antes de terminar la comunicación. TCP permite que los paquetes lleguen al receptor con el mismo orden con el que fueron emitidos desde el transmisor y chequea si se han producido errores en la comunicación. Si el emisor del paquete no recibe confirmación de que éste ha llegado a su destino, vuelve a enviarlos. Es por estas

características del protocolo que no es apropiado para aplicaciones que necesiten velocidad, pues tanto la sincronización, la desincronización y la comprobación de errores ocupan parte significativa del régimen binario de la comunicación. En cambio, sí resulta útil en casos donde la fiabilidad y la integridad de los mensajes son importantes. En la cabecera de dicho protocolo existe un identificador de secuencia (*sequence number*) de 4 bytes (ver Figura 23). Este campo es compartido entre los dos extremos de la comunicación, lo que facilita en gran medida la identificación de los paquetes.

Protocolos basados en TCP con Telnet, para el manejo remoto de terminales; FTP, para la transmisión de ficheros; SMTP, para el envío de correo electrónicos; HTTP, la base de las páginas web, o RMC, para la ejecución de procesos remotos.

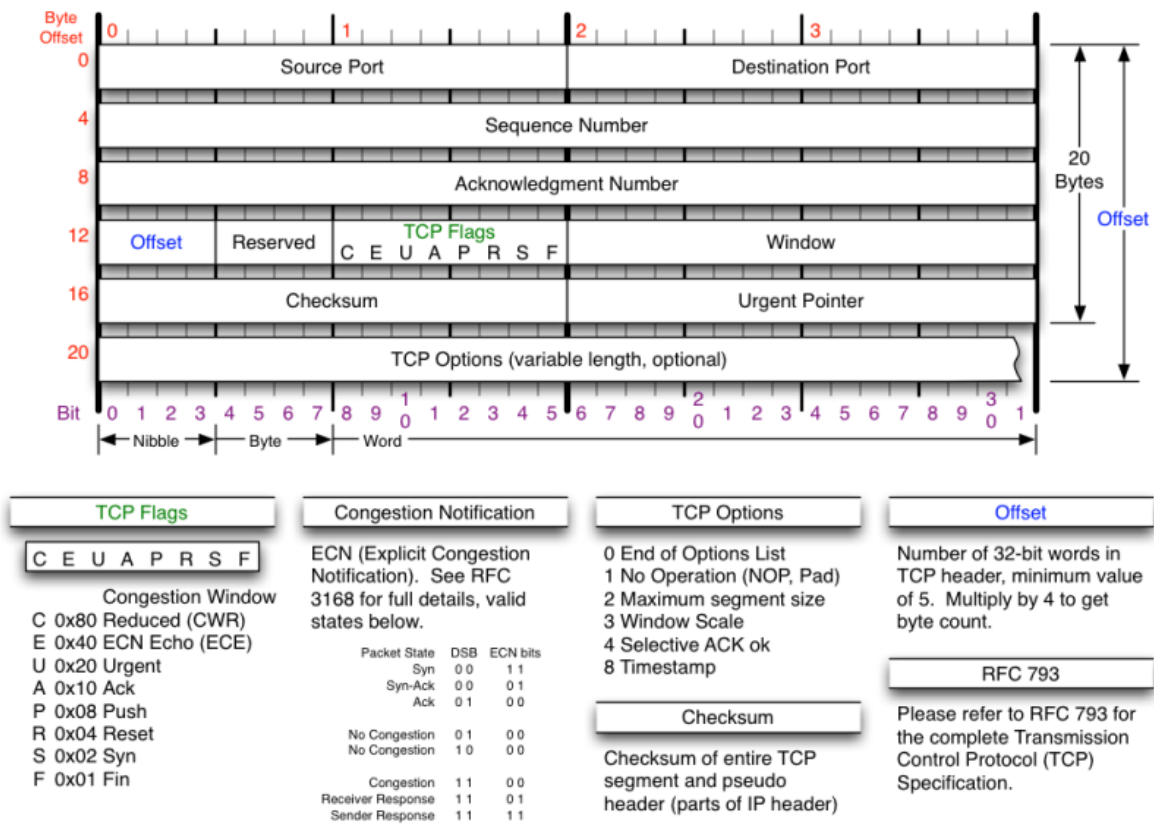


Figura 23. Estructura de la cabecera TCP

Por el contrario, el otro gran protocolo de la capa de transporte, UDP, ofrece un servicio no fiable y no orientado a conexión basado en la filosofía *best-effort*, consistente en que los nodos reciben el mejor servicio posible en cada momento. Este “mejor servicio posible”, variante según las circunstancias, depende estrechamente de la velocidad de envío de datos por parte del emisor, y no garantiza la llegada de los paquetes (datagramas) a su destino, pues en caso de saturación de la capacidad los datagramas se pierden. En definitiva, en UDP los paquetes simplemente se envían, pero no se comprueba si llegan a su destino. Del mismo modo, pueden llegar desordenados o duplicados. Las menores prestaciones de UDP reducen su complejidad, y esto redundará en una reducción del tamaño de su cabecera en comparación con TCP. Por esta simplicidad, UDP se utiliza en aplicaciones donde la velocidad es más importante que las pérdidas, tales como juegos *on line* o vídeos en *streaming*. Todas estas aplicaciones tienen como características comunes que:

- Tienen sus propios mecanismos para conexiones, control de flujo y comprobación de errores.
- Alivian la carga de retransmisiones.
- Se basan en un modelo de petición – respuesta.

Protocolos basados en UDP son DNS, para la resolución de nombres en internet, o BOOTP, este último utilizado para el arranque remoto de sistemas.

Como puede verse en la Figura 24 la cabecera del protocolo UDP sólo consta de campos indicadores del puerto origen y destino en las comunicaciones, la longitud de los datos y un campo de seguridad o *checksum* para comprobar que estos no son modificados durante su trayecto, siendo además este último optativo. No se cuenta, por tanto, con un identificador de trama como en el caso de TCP. Este inconveniente dificulta la identificación de los paquetes enviados y recibidos, rémora que se tendrá en consideración a la hora de medir la calidad de las conexiones.

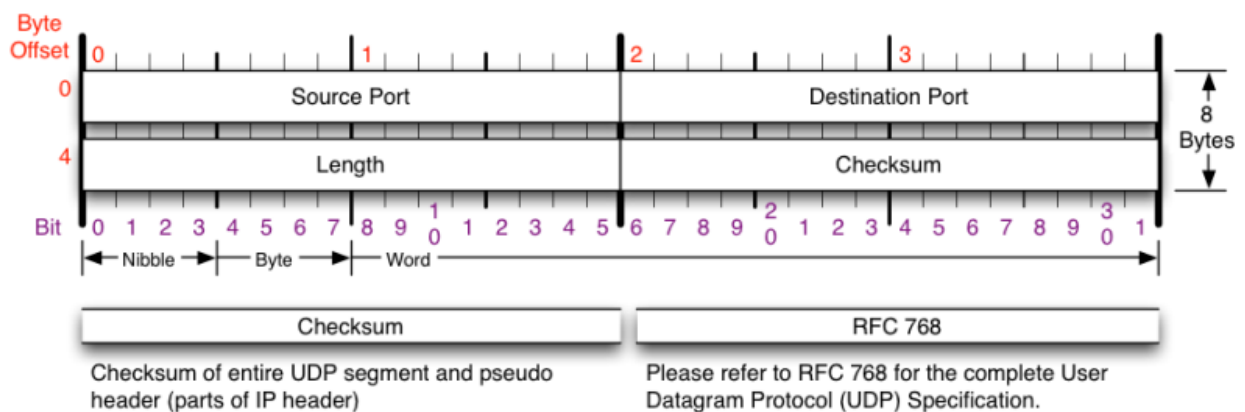


Figura 24. Estructura de la cabecera UDP

### 3.3.4. Wireshark: gestión de capturas de tráfico

Para analizar los datos relativos a la calidad de las comunicaciones es necesario realizar anteriormente capturas del tráfico que se produce entre las aplicaciones cliente y servidor. Para ello se utilizará la herramienta tshark. Se trata de un analizador de protocolos de red que, tras ejecutarse desde consola (previa instalación en Raspbian), originará un fichero de captura con extensión .pcap que procesará y registrará en tiempo real todo el tráfico que se esté dando. Una vez finalizado el proceso de comunicación, el fichero de captura podrá abrirse con Wireshark.

Wireshark otorga gran manejabilidad en el tratamiento de la captura previamente guardada. Una de las potencialidades a destacar es la posibilidad de filtrar los paquetes según las necesidades del usuario. Se distinguen dos tipos de filtro: los de captura (*capture filters*) y visualización (*display filters*). Como su propio nombre indica, los de captura indican aquellos paquetes que deben registrarse en el fichero de captura, y por tanto deben indicarse con anterioridad al comienzo de la misma. Por otro lado, los filtros de visualización se aplican sobre el fichero de captura ya guardado, y hacen que se muestren sólo una parte de los paquetes que han sido capturados. Esto, sin embargo, no significa que los paquetes que no concuerden con el filtro aplicado sean eliminados: simplemente se ocultan mientras el filtro esté activo. A modo de ejemplo, en la Figura 25 se muestra un filtro de visualización que atañe a los paquetes en los que el protocolo de transporte utilizado sea

TCP y que hayan sido enviados desde la dirección de red 192.168.1.137, que fue asignada con anterioridad a la interfaz inalámbrica de una de las RPi.

tcp && ip.src==192.168.1.137					
No.	Time	Source	Destination	Protocol	Length
53	20:50:39.530910	192.168.1.137	90.106.186.114	TCP	66
54	20:50:39.530910	192.168.1.137	179.254.27.210	TCP	66
55	20:50:39.531188	192.168.1.137	38.88.33.58	TCP	66
56	20:50:39.531188	192.168.1.137	50.200.49.83	TCP	66
57	20:50:39.531204	192.168.1.137	95.105.187.2	TCP	66
58	20:50:39.531527	192.168.1.137	88.132.147.185	TCP	66

Figura 25. Filtro de visualización en Wireshark

A continuación se indican algunos filtros que han resultado útiles para la obtención de los parámetros de las comunicaciones:

- `tcp && ip.src==192.168.1.137 && ip.dst==192.168.1.138 && (frame.time >= "Sep 2, 2017 21:55:54") && (frame.time <= "Sep 2, 2017 21:57:48") && frame.len!=66`. Este filtro visualiza los paquetes TCP con datos "útiles" (es decir, los mensajes que se han mandado) enviados desde el cliente (192.168.1.137) hasta el servidor (192.168.1.138) en un intervalo de tiempo dado, entre las 21:55:54 y las 21:57:48 del 2 de septiembre.
- `udp && !ntp && ip.src==192.168.1.137 && ip.dst==192.168.1.138 && frame.time < "Sep 2, 2017 21:46:40.9638000"`. Muestra los paquetes UDP (sin incluir al protocolo NTP de sincronización horaria) enviados desde el cliente (192.168.1.137) hasta el servidor (192.168.1.138) antes de las 21:46:40 del 2 de septiembre.
- `tcp && ip.src==192.168.1.137 && ip.dst==192.168.1.138 && !ssh && tcp.dstport==9999 && tcp.len!=0 && frame.time>"Sep 7, 2017 17:47:10.805947" && frame.time<"Sep 7, 2017 17:48:28.805947"`. Con este filtro se visualizan los segmentos TCP enviados desde el cliente hasta el servidor, siendo el puerto de destino el 9999 (que es el que se ha preconfigurado en el programa clientetcp.py). Se han excluido las tramas SSH (protocolo que se sirve de TCP), así como los segmentos que no tengan carga útil (es decir, aquellos que sirven para sincronizar los extremos de la conexión o bien los asentimientos ACK).
- `tcp && ip.src==192.168.1.137 && ip.dst==192.168.1.138 && !ssh && tcp.dstport==9999 && tcp.len!=0 && frame.time>"Sep 7, 2017 17:47:10.805947" && frame.time>"Sep 7, 2017 17:48:28.805947" && !tcp.analysis.retransmission`. Filtro que actúa del mismo modo que el anterior, pero con la condición de no mostrar retransmisiones de paquetes.
- `tcp && ip.src==192.168.1.137 && ip.dst==192.168.1.138 && !ssh && tcp.dstport==9999 && tcp.len!=0 && frame.time>"Sep 7, 2017 17:47:10.805947" && frame.time>"Sep 7, 2017 17:48:28.805947" && !tcp.analysis.retransmission && !tcp.analysis.duplicate_ack_frame`. Con la misma función que el último, añadiendo además la eliminación de asentimientos duplicados.

Otra prestación de Wireshark de la que se sacará provecho es aquella que permite exportar datos en un archivo CSV. Un archivo CSV (*comma separated value*, por sus siglas en inglés) es una representación en texto plano de datos tabulares, que quedarán separados por comas. Wireshark exportará al archivo CSV aquellos campos que se muestren en la lista de paquetes, previa aplicación de los filtros de visualización mencionados con anterioridad. La ventaja de estos ficheros es que Python provee de módulos dedicados a su procesamiento de forma sencilla, de forma que, por ejemplo, puedan pasarse a un array determinadas columnas del archivo CSV mediante un algoritmo simple.

La interfaz de Wireshark se muestra en la Figura 26. Como se ve, se divide en tres zonas bien diferenciadas: una correspondiente a la lista de paquetes, otra a detalles de cada paquete reproduciendo la estructura del modelo OSI (así se muestran parámetros referidos al medio físico, posteriormente Ethernet, Internet Protocol, Transmission Control Protocol, etc.) y, por último, la parte inferior muestra la información en bytes de cada paquete, es decir, el paquete en sí como una secuencia de unos y ceros. Esta última sección resulta útil cuando se transmiten cadenas de texto y se quiere ver cómo son codificadas en el paquete. Sin embargo, las secciones que más se utilizarán serán las dos primeras. Las columnas que se muestran (en este caso número de paquete, tiempo de captura, dirección de red origen y destino, etcetera) son aquellas cuyos campos se pasarán al fichero CSV una vez se seleccione la opción de exportar. Estas columnas, no obstante, pueden añadirse o eliminarse al



gusto, es decir, es posible que la configuración por defecto de Wireshark no muestre en la sección superior ciertas columnas que resulten de interés. En la sección de detalles de cada paquete se muestran una serie de campos que, en caso de interesar, son susceptibles de pasarse a la sección superior para que así se muestren en el archivo CSV. Esto facilitará en gran medida la labor de caracterización de las comunicaciones inalámbricas, pues permitirá que a partir de un único fichero de captura éste pueda manipularse de la forma que se ha indicado antes para mostrar en el archivo CSV los datos que resulten pertinentes para cumplir con esta labor.

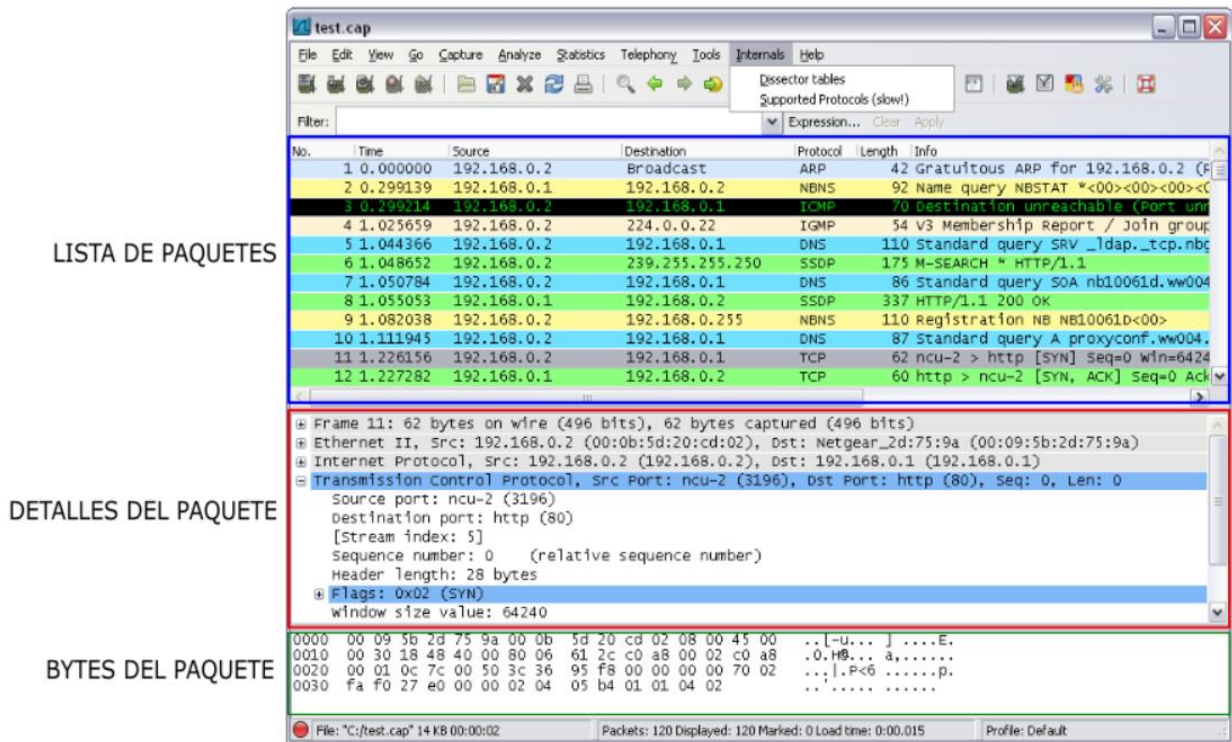


Figura 26. Interfaz de Wireshark

Resulta importante reseñar que, para conseguir que las dos RPi tengan una visión integral de lo que está ocurriendo en la red, las capturas deben comenzar en el momento en que se enciendan los ordenadores, y a partir de ahí esperar a que se registren transmisiones en la red. Esto se consigue introduciendo los comandos necesarios para realizar la captura en un script que se localice en el directorio `~/config/autostart`. Una operación similar ya se hizo para tener siempre listo el servidor VNC, sin necesidad de iniciarlo manualmente.

```
cd /
cd /etc/wireshark
sudo tshark -i 2 -w $(date +%Y_%m_%d_%L%M_%p).pcap
sudo chmod -R 0755 /etc/wireshark
```

Figura 27. Script que realiza la captura del tráfico

El script a utilizar se muestra en la Figura 27. Las líneas más relevantes del mismo son la tercera y la cuarta. En la línea tercera se ejecuta la captura del tráfico que se produce en la interfaz inalámbrica de la RPi. Obsérvese que una de las opciones que acompañan al comando es `-i 2`; con ella se indica la interfaz cuyo tráfico queremos capturar, en este caso la inalámbrica. Para conocer la correspondencia entre números e interfaces basta con ejecutar el comando `tshark -D` (ver Figura 28).

```
pi@raspberrypi:~ $ tshark -D
1. eth0
2. wlan0
3. any
4. lo (Loopback)
5. nflog
6. nfqueue
7. usbmon1
```

Figura 28. Relación de interfaces cuyo tráfico puede capturarse con tshark

Por otro lado, el fichero en el que se escriben los datos referentes a la captura se especifica a continuación del parámetro “-w”. Al tratarse de un script que se ejecuta al activar la RPi, utilizar un nombre predeterminado para la captura ocasionaría que el fichero se sobrescribiese cada vez que se enciende el ordenador, y por tanto se perdería la información conseguida en ocasiones anteriores. Es por ello que el nombre del fichero con extensión .pcap será el de la fecha en el que se ha ejecutado el comando. En este caso la única pérdida de captura se daría si se ejecutan dos durante un mismo minuto (verbigracia, si se activa una captura el día 6 de junio de 2017 a las 18:50 horas, la siguiente sólo podría realizarse, como mínimo, el 6 de junio de 2017 a las 18:51 horas) para no ocasionar la sobrescritura de datos. Se ve, además, que en esta primera fase de captura no se utiliza filtro alguno. La disección de los datos, por tanto, se aplaza a la fase de postprocesamiento.

En lo que respecta a la cuarta línea, ésta simplemente proporciona permisos en el directorio donde éstas se guardan, con vistas a poder abrirlas y manipularlas en Wireshark posteriormente. Si este comando no fuera suficiente, se recomienda cambiar propietario y permisos de todos los archivos contenidos en la carpeta */etc/wireshark* mediante los comandos *sudo chown pi:pi \*.pcap* y *sudo chmod +x \*.pcap*.

### 3.3.5. Programas de procesamiento de datos relativos a las características de las comunicaciones inalámbricas

Los parámetros de calidad que se pretenden caracterizar tras el proceso de comunicación son tres:

- tasa de recepción de paquetes, definida como el cociente entre los paquetes que llegan al servidor y los que han sido enviados desde el cliente,
- rendimiento o *throughput*, definido como la velocidad a la que llegan los datos al servidor,
- retraso, es decir, el lapso de tiempo que transcurre desde el paquete se envía desde un nodo origen hasta el nodo destino.

Llegados a este punto ya se cuenta con un fichero de captura de tráfico. En este apartado se explicará de forma somera el algoritmo aplicado para obtener cada uno de los datos de interés en la caracterización de las comunicaciones (tasa de recepción, throughput y retardo). Por si resultara de interés, en la sección “Códigos utilizados”, al final de este documento, se reproducen los ficheros Python utilizados en el desarrollo de este proyecto.

Para calcular la tasa de recepción de paquetes se recurrirá a dos metodologías distintas según el protocolo de transporte que se esté utilizando, TCP o UDP. Esta disyuntiva surge de la ya mencionada con anterioridad especificación del protocolo TCP, que incluye un identificador de secuencia para cada paquete gestionado. En este caso, para hallar esta tasa de paquetes transmitidos con éxito entre cliente y servidor basta con incluir en dos arrays los números de secuencia procesados por Wireshark en cada extremo de la comunicación, para a continuación ver cuántos elementos son comunes a ambos arrays en relación con el número de elementos transmitidos desde el otro nodo que toma parte en la comunicación. Naturalmente, estos números de secuencia deben constar en el archivo CSV. Dividiendo ambas cifras se obtiene un coeficiente que, multiplicado por cien, resulta la tasa de recepción de paquetes.

En lo que se refiere al throughput o regimen binario de la comunicación, el protocolo de transporte que se esté usando es indiferente, pues para cada extremo de la comunicación simplemente se trata de introducir en una lista el tamaño en bytes de los paquetes transmitidos, y los tiempos relativos de captura (es decir, en relación

con el comienzo de la misma) de los paquetes en otro. Sumando todos los elementos del primer array, el de tamaños de paquetes, y dividiendo la cifra obtenida por el último elemento del array de tiempos relativos se halla el caudal de transmisión. Análogamente a lo indicado en el párrafo anterior, en este caso serán los tiempos relativos de captura y los tamaños del paquete los campos que, entre otros, deberán constar en el archivo CSV.

Por ultimo, el retardo medio en la comunicación ha sido sin duda el parámetro que más quebraderos de cabeza ha ocasionado a la hora de ser calculado, pues implica vérselas de bruces con las limitaciones de RPi e intentar aportar soluciones que se sobrepongan a estas debilidades de los dispositivos con los que se ha trabajado. Para calcular los retrasos resulta imprescindible contar con una referencia temporal que permita visualizar cuánto tiempo transcurre desde que un paquete se envía hasta que se recibe. Esta referencia temporal solo está disponible en RPi si hay una conexión a internet que aporte esta información. El remedio ideado ha sido ya desarrollado en apartados anteriores: al disponer en el experimento de una única RPi conectada a internet, la única solución que implica que ambas tarjetas estén sincronizadas pasa por que esta RPi conectada a internet actúe por un lado como cliente (es decir, actualizando su información horaria en concordancia con la información que reciba de la red) y por otro, como servidor, proporcionando esta información mediante la utilización del protocolo NTP y el aprovechamiento de la red ad-hoc que servirá de canal de comunicación entre ambas RPi.

Existe, no obstante, otro método para calcular el retardo medio de las comunicaciones, si bien requiere de un preprocesamiento del archivo CSV. Wireshark ofrece la opción de fijar los parámetros temporales de un paquete como referencia respecto a los otros, de manera que si, por ejemplo, un paquete llega a los cinco segundos de iniciarse la captura y el siguiente lo hace a los siete segundos, fijando el primer paquete como referencia, el tiempo de llegada del siguiente constará como dos segundos. Como se muestra en la Figura 29, esto se realiza seleccionando la opción "Set/Unset Time Reference" sobre cualquier paquete.

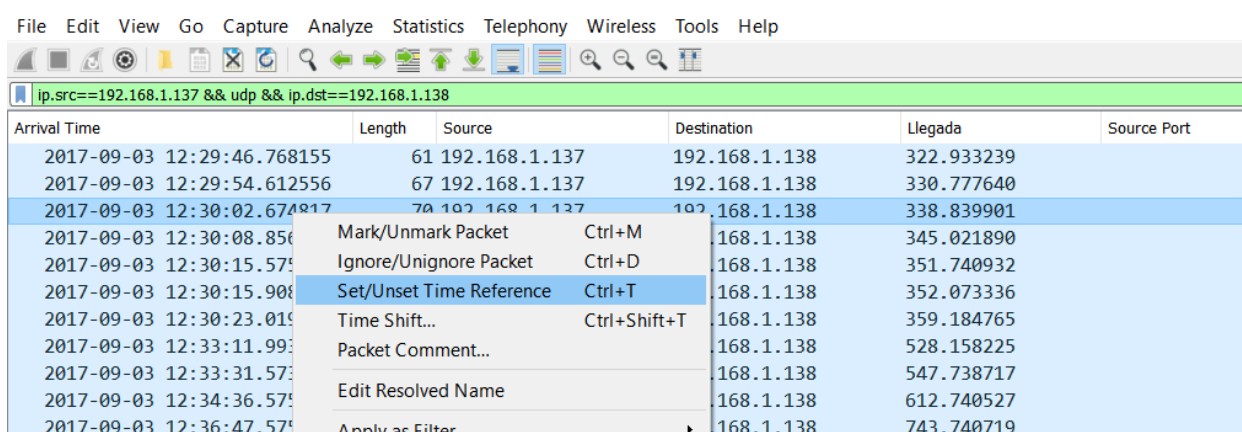


Figura 29. Opción para fijar una referencia temporal en Wireshark

Por ende, utilizando un paquete como referencia, pueden calcularse el retardo en la transmisión del resto de los paquetes. Se trata de una opción a considerar, vista la inconsistencia del protocolo NTP, fenómeno que a grandes distancias entre las RPi puede agravarse, imposibilitando la sincronización.

### 3.3.6. Ejecución de tareas: SSH

Una de las peculiaridades del proyecto es que no se tiene acceso físico a una de las Raspberry Pi durante el desarrollo de las pruebas, pues este se encontrará a una distancia lejana (en el caso de 2D) o, incluso, a alturas distancias y alturas elevadas cuando la RPi esté situada en el dron y se esté modelando la calidad de las comunicaciones 3D.

El protocolo *Secure Shell* – más conocido por sus siglas SSH- es un método para acceder a equipos remotos ofreciendo una conexión, a través del puerto 22 TCP, segura, íntegra y encriptada (ver Figura 30). Además, permite:

- La transferencia de ficheros.
- La ejecución de comandos en el equipo remoto.
- La gestión de la infraestructura de red.

Es por todas las razones aquí presentadas que SSH será de mucha utilidad para acceder al sistema operativo de la Raspberry Pi que está colocada sobre el dron.

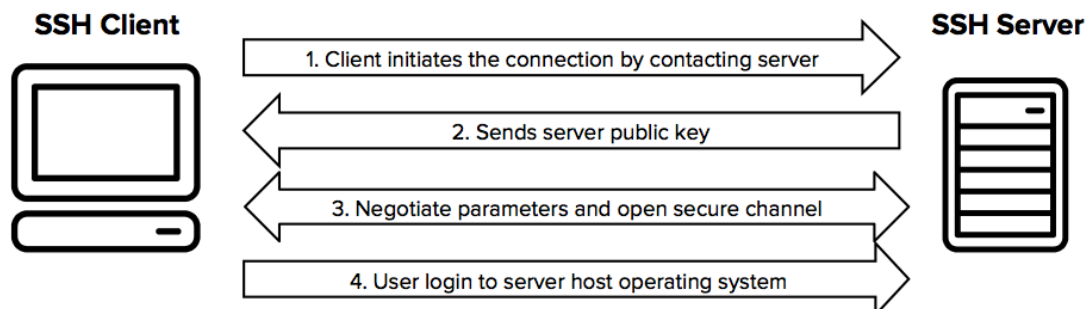


Figura 30. Autenticación basada en el protocolo SSH

Una vez que la RPi servidor esté dispuesta sobre el UAV, y con este volando, puede accederse a ella mediante una conexión ssh. Para ello, basta con ejecutar desde el cliente el comando `ssh 192.168.1.138` e introducir la contraseña `raspberrypi`. Desde ese momento tendremos disponible la consola de la RPi que está colocada en el UAV, y por ende podremos ordenar desde la estación de tierra tanto el arranque remoto de los servidores TCP y UDP, como el de la captura en el lado del servidor.

El programa `scp` (*Secure Copy Protocol*) hace uso de SSH para la transferencia de archivos entre distintos equipos. Se aprovechará el uso de este para transferir los ficheros de capturas en el lado del cliente hacia el servidor, y de esta forma tener una copia de seguridad de estos archivos ante un eventual fallo de cualquiera de las Raspberry Pi. La sintaxis de este comando es la siguiente: `scp /ruta/del/fichero usuario@ip_host_remoto:/directorio/destino/`.

# 4 BANCO DE PRUEBAS

*It doesn't matter how beautiful your theory is, it doesn't matter how smart you are.  
If it doesn't agree with experiment, it's wrong.*

*- Richard Feynman, premio Nobel de Física -*

Como se mencionó con anterioridad, las pruebas a realizar tienen como objetivo obtener tres métricas de las comunicaciones:

- Tasa de recepción de paquetes, o PRR por sus siglas en inglés. Define el ratio de paquetes recibidos con éxito en relación con aquellos que fueron enviados. Tasas altas suponen una mayor fiabilidad en las transmisiones y calidad de la red.
- Caudal o tasa binaria de recepción, definido como el número de bits (o unidades derivadas) transmitidos a través de la red por unidad de tiempo. Es, por tanto, la velocidad de transferencia de datos del sistema de comunicaciones. Obviamente, valores altos son preferibles para este parámetro.
- Retardo o lapso de tiempo medio que transcurre entre que se envía un paquete y su recepción en el otro extremo de la comunicación. Se trata de un parámetro íntimamente relacionado con la tasa binaria, e inversamente proporcional a él. Por tanto, un sistema de comunicaciones de calidad deberá presentar valores bajos de tiempos.

Las pruebas a realizar se dividirán en estáticas y dinámicas:

- Estáticas:
  - Prueba E.1.: Comunicación entre dos RPi en tierra utilizando el protocolo UDP, con una de ellas recurriendo a un dongle Wi-Fi y otra al módulo Wi-Fi integrado.
  - Prueba E.2.: Comunicación entre dos RPi en tierra utilizando el protocolo TCP, con una de ellas recurriendo a un dongle Wi-Fi y otra al módulo Wi-Fi integrado.
  - Prueba E.3.: Comunicación entre dos RPi en tierra utilizando el protocolo UDP, y ambas recurriendo al módulo Wi-Fi integrado.
  - Prueba E.4.: Comunicación entre dos RPi en tierra utilizando el protocolo TCP, y ambas recurriendo al módulo Wi-Fi integrado.

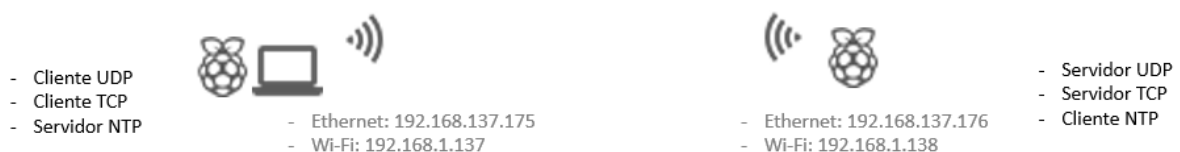


Figura 31. Escenario de las pruebas estáticas

- Dinámicas:
  - Prueba D.1.: Comunicación entre RPi configurada como estación de control en tierra (GCS) y RPi en UAV utilizando el protocolo UDP, con la RPi de GCS recurriendo a un dongle Wi-Fi y la de UAV al módulo Wi-Fi integrado.
  - Prueba D.2.: Comunicación entre RPi configurada como estación de control en tierra (GCS) y

- RPi en UAV utilizando el protocolo TCP, con la RPi de GCS recurriendo a un dongle Wi-Fi y la de UAV al módulo Wi-Fi integrado.
- Prueba D.3.: Comunicación entre RPi configurada como estación de control en tierra (GCS) y RPi en UAV utilizando el protocolo UDP, con ambas RPi recurriendo al módulo Wi-Fi integrado.
  - Prueba D.4.: Comunicación entre RPi configurada como estación de control en tierra (GCS) y RPi en UAV utilizando el protocolo TCP, con ambas RPi recurriendo al módulo Wi-Fi integrado.

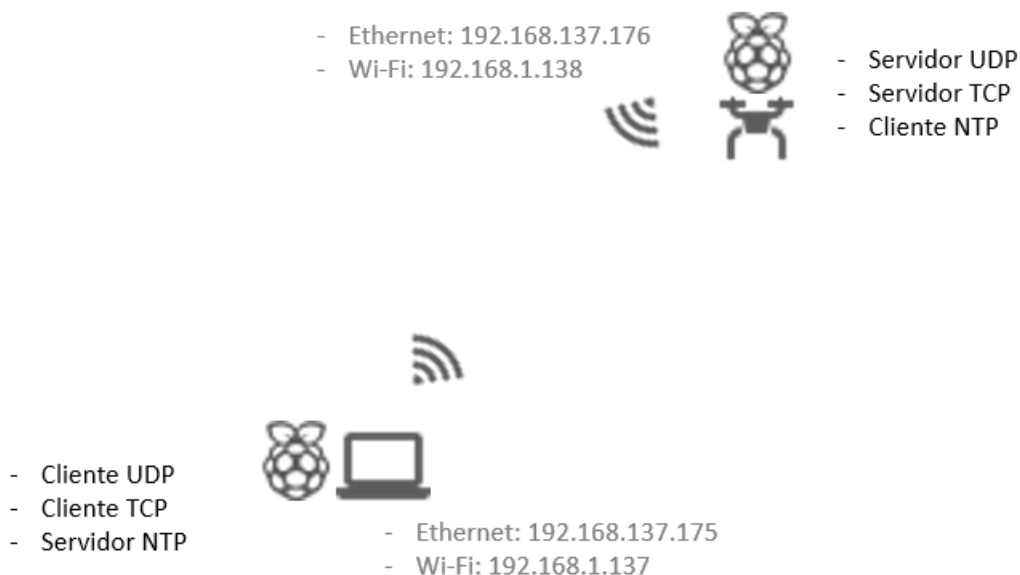


Figura 32. Escenario de las pruebas dinámicas

Cada uno de estos casos de prueba se realizarán con distintas distancias entre los dos dispositivos conectados para así comprobar las consecuencias que la separación física entre las Raspberry Pi tienen sobre la calidad de la señal.

Para tratar de reproducir ambos grupos de pruebas bajo las mismas condiciones, la conexión a internet del ordenador (que está conectado por Ethernet a la Raspberry Pi que actúa como cliente enviando mensajes TCP y UDP) se consigue haciendo que un *smartphone* con sistema operativo Android se comporte como proveedor de señal. Además, los mensajes que se mandan en cada una de las pruebas deben de ser los mismos en todos los casos de la *suite* de prueba, pues es así como se logrará apreciar las diferencias en la calidad de las comunicaciones derivadas de las distintas configuraciones. En concreto, el texto enviado corresponde a un fragmento del poema *Happiness*, de Raymond Carver, y figura en el apartado “Mensaje transmitido”, adjunto en las páginas finales de este documento.

Las pruebas dinámicas fueron realizadas en las instalaciones de la Escuela Técnica Superior de Ingeniería de Sevilla el día 7 de septiembre de 2017. De estos tests puede esperarse un comportamiento distinto al observado en las pruebas estáticas; principalmente por dos motivos:

- La existencia de un entorno inestable, no sólo por los fenómenos inherentes a un espacio abierto: vulnerabilidad al tiempo atmosférico, presencia de obstáculos, interferencias electromagnéticas (EMI), etcétera.
- La propia naturaleza del dron [28], que se muestra en la Figura 43. Se trata de un quadrotor con un chasis de setenta centímetros entre extremos de los brazos, dispuestos en línea recta dos a dos, y peso aproximado de 1,5 kilogramos. Por la gran cantidad de componentes electrónicos y circuitería que integra, entre ellos placa de control, receptor de radio, indicador de voltaje de batería, PDB (*Power Distribution Board*) y batería, hacen que el fenómeno de interferencia electromagnética no pueda

despreciarse [29]. Por si no bastase, la Raspberry Pi que se coloca en el UAV contribuye a acrecentar este efecto que empeora notablemente la calidad de la señal Wi-Fi.

*Todos los gráficos mostrados en este apartado están basados en librerías de Python, como matplotlib. En el apéndice de Códigos utilizados, al final de este documento, se consignan los códigos que hicieron uso de esas librerías.*

## 4.1. Pruebas E.1 y E.2: UDP y TCP con dongle

En las pruebas estáticas las dos Raspberry Pi se encuentran en posiciones fijas durante la transmisión de los mensajes, dentro de una vivienda, con lo que ello implica a la hora de descartar fenómenos atmosféricos que pudieran influir en las comunicaciones.

```
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:10:01: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:10:06: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:10:13: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:10:19: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:10:25: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:10:35: walking up the road
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:10:42: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:10:49: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:10:55: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:11:03: They are so happy
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:11:10: they aren't saying anything, these boys.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:11:50: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:11:55: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:12:03: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:12:10: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:12:16: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:12:25: walking up the road
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:12:31: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:12:37: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:12:43: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:12:49: They are so happy
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:12:55: they aren't saying anything, these boys.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:13:50: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:13:57: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:14:03: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:14:10: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:14:16: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:14:22: walking up the road
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:14:28: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:14:34: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:14:39: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:14:46: They are so happy
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:14:54: they aren't saying anything, these boys.
```

Figura 33. Fichero log correspondiente a la prueba E.1

```

Esperando conexiones...
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:02:59: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:03:05: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:03:12: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:03:18: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:03:25: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:03:31: walking up the road
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:03:37: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:03:42: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:03:48: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:03:54: They are so happy
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:04:01: they aren't saying anything, these boys.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:04:43: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:04:49: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:04:54: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:05:00: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:05:06: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:05:12: walking up the road
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:05:18: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:05:24: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:05:30: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:05:37: They are so happy
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:05:43: they aren't saying anything, these boys.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:06:35: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:06:40: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:06:45: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:06:51: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:06:57: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:07:02: walking up the road
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:07:08: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:07:15: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:07:20: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:07:26: They are so happy
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:07:33: they aren't saying anything, these boys.
Cerrando

```

Figura 34. Fichero log correspondiente a la prueba E.2

En las siguientes gráficas se comprobarán las diferencias en comportamiento en función de la distancia entre Raspberry Pi y protocolo en lo que refiere a la tasa de recepción de paquetes, la velocidad de transmisión y el retardo medio. De la Figura 35 se desprende que la tasa de recepción de paquetes es máxima a una distancia de seis metros, con independencia del protocolo utilizado. Los resultados son, así, plenamente satisfactorios.

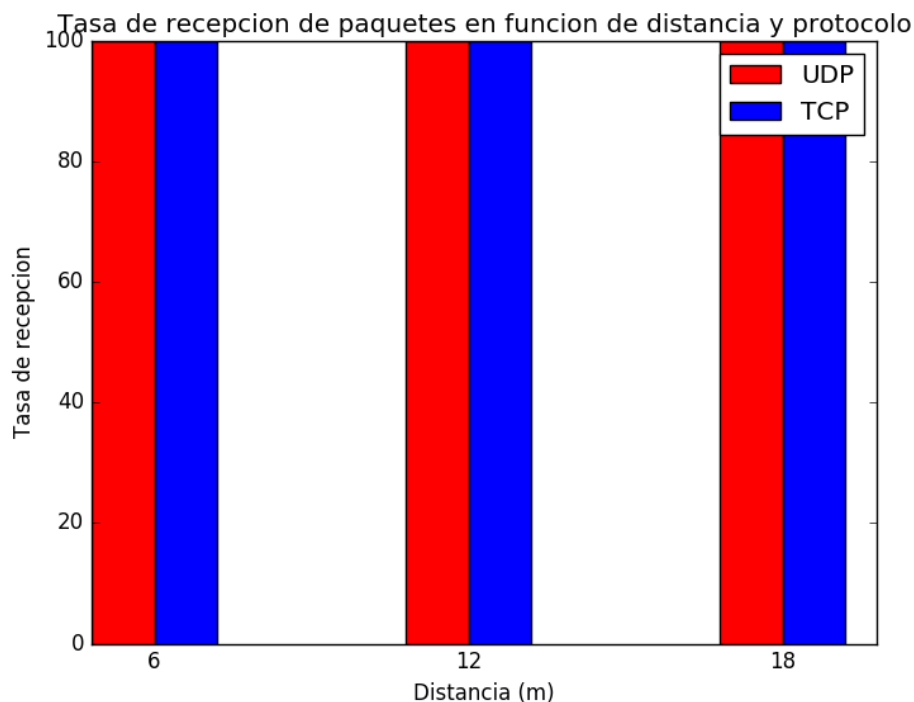


Figura 35. Pruebas E.1 y E.2: Tasa de recepción de paquetes en función de la distancia y el protocolo de transporte utilizado



En lo que respecta a la tasa binaria, se mantiene estable para distintas distancias, si bien en TCP este caudal es mayor. Sin embargo, esto no debe llevar a equívoco: téngase en cuenta que la carga de cabecera del protocolo TCP es considerablemente mayor que la UDP, y por tanto, la mayor tasa de TCP se debe en gran parte a la transmisión de datos ajenos a los mensajes enviados. En cuanto a los retardos, aumentan sensiblemente en TCP para la distancia máxima de dieciocho metros.

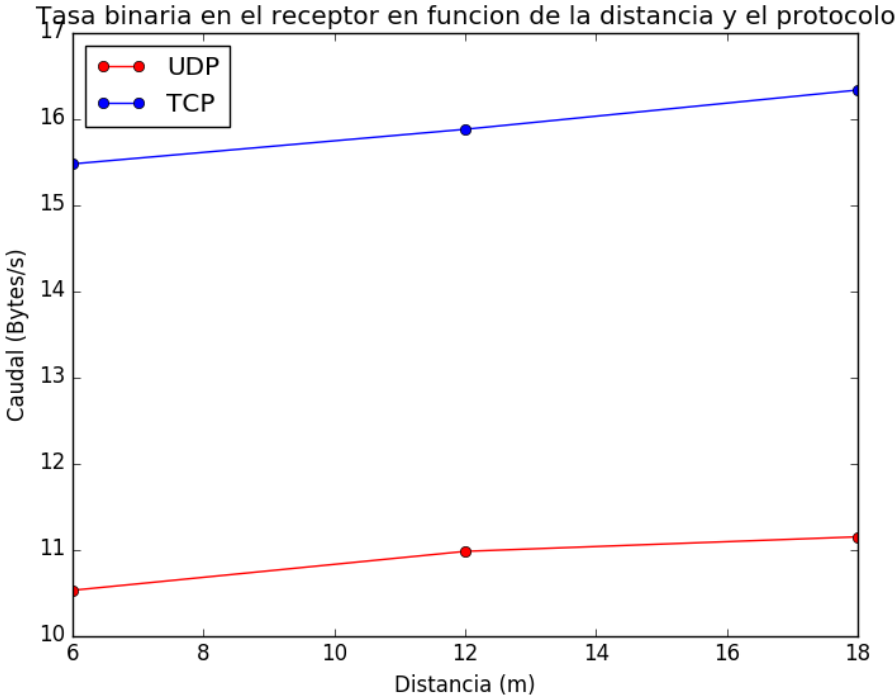


Figura 37. Pruebas E.1 y E.2: Tasa binaria de recepción de paquetes en función de la distancia y el protocolo de transporte utilizado

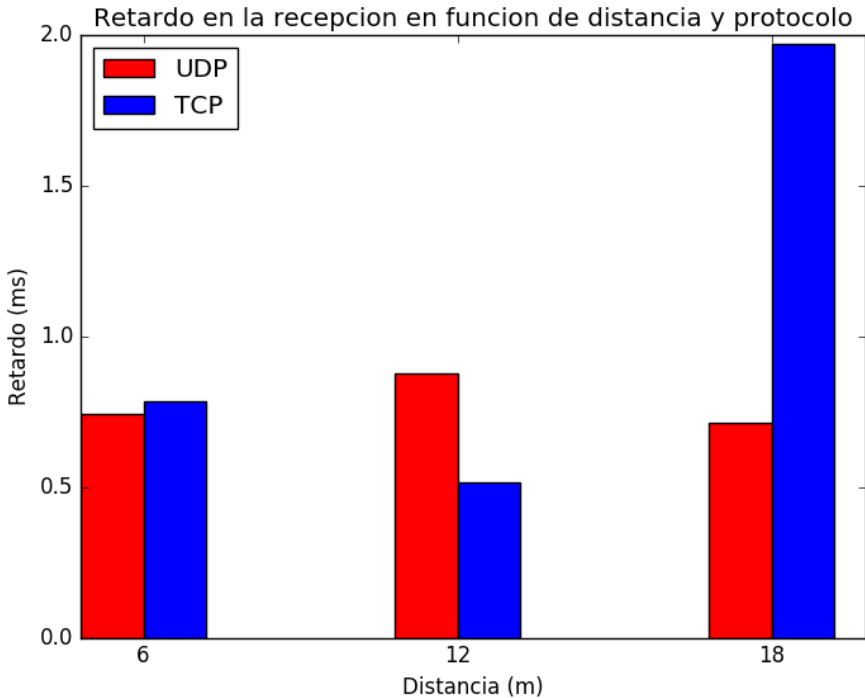


Figura 36. Pruebas E.1 y E.2: Retardo en la recepción de paquetes en función de la distancia y el protocolo de transporte utilizado

## 4.2. Pruebas E.3 y E.4: UDP y TCP con transmisor integrado

Estas pruebas testarán la calidad de las comunicaciones entre RPi en posiciones fijas cuando estas se valen del módulo Wi-Fi integrado.

```
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:42:12: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:42:19: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:42:25: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:42:32: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:42:38: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:42:45: walking up the road
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:42:52: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:42:58: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:43:05: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:43:11: They are so happy
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:43:18: they aren't saying anything, these boys.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:44:16: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:44:21: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:44:28: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:44:34: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:44:41: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:44:47: walking up the road
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:44:53: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:44:59: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:45:05: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:45:18: They are so happy
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:45:24: they aren't saying anything, these boys.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:46:16: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:46:22: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:46:31: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:46:37: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:46:43: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:46:50: walking up the road
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:46:57: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:47:03: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:47:10: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:47:17: They are so happy
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:47:23: they aren't saying anything, these boys.
```

Figura 39. Fichero log correspondiente a la prueba E.3

```
Esperando conexiones...
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:53:42: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:53:48: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:53:55: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:54:02: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:54:10: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:54:17: walking up the road
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:54:23: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:54:30: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:54:36: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:54:44: They are so happy
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:54:51: they aren't saying anything, these boys.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:55:44: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:55:51: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:55:57: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:56:03: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:56:10: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:56:22: walking up the road
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:56:27: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:56:34: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:56:40: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:56:47: They are so happy
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:56:54: they aren't saying anything, these boys.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:57:45: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:57:52: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:57:58: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:58:04: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:58:17: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:58:22: walking up the road
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:58:28: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:58:34: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:58:39: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:58:46: They are so happy
Mensaje recibido de 192.168.1.137 el día 11-09-2017 a las 19:58:52: they aren't saying anything, these boys.
Cerrando
```

Figura 38. Fichero log correspondiente a la prueba E.4

Fijando ahora el foco sobre el modulo Wi-Fi integrado, la Figura 40 sigue la tónica de los resultados de las pruebas E.1 y E.2, con tasas de recepción máximas en todos los casos. El caudal de transmisión (ver Figura 41) se mantiene estable, siendo las divergencias encontradas achacables a fenómenos aleatorios de los cuales no se pueden obtener conclusiones en cuanto a calidad de las comunicaciones. Los retardos (ver Figura 41) resultan estables en TCP, no así en UDP. De todos modos, en ningún caso toman valores prohibitivos que inhabiliten el uso del sistema de comunicaciones para aplicaciones cotidianas del UAV.

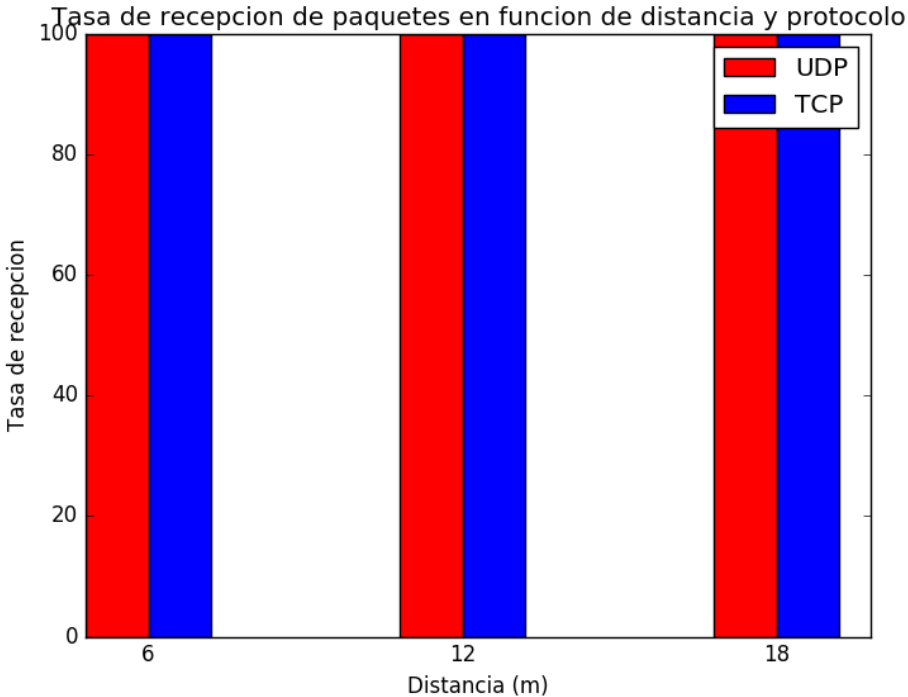


Figura 40. Pruebas E.3 y E.4: Tasa de recepción de paquetes en función de la distancia y el protocolo utilizado

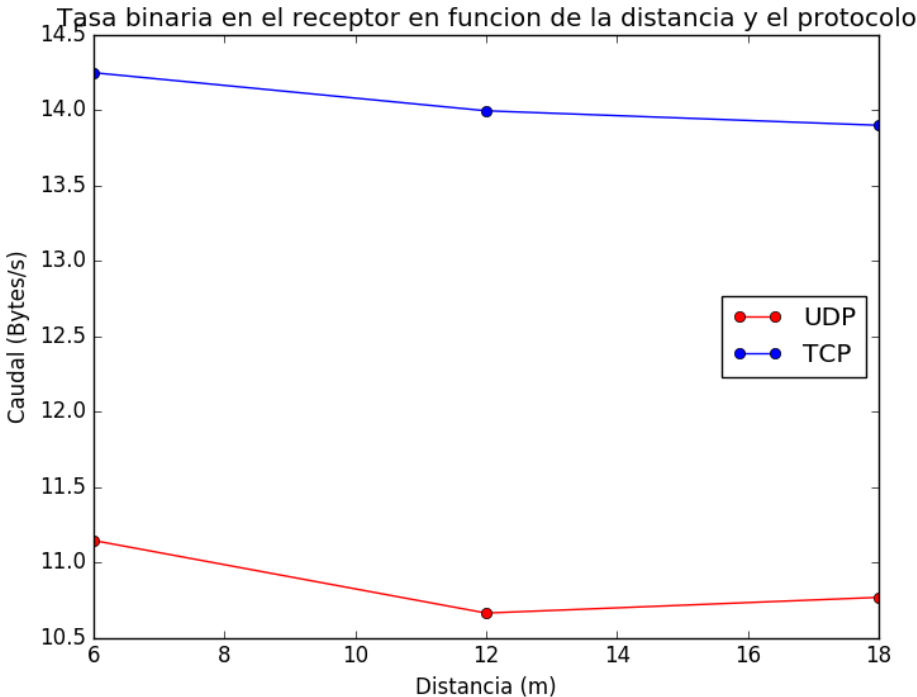


Figura 41. Tasa binaria de recepción de paquetes en función de la distancia y el protocolo de transporte utilizado

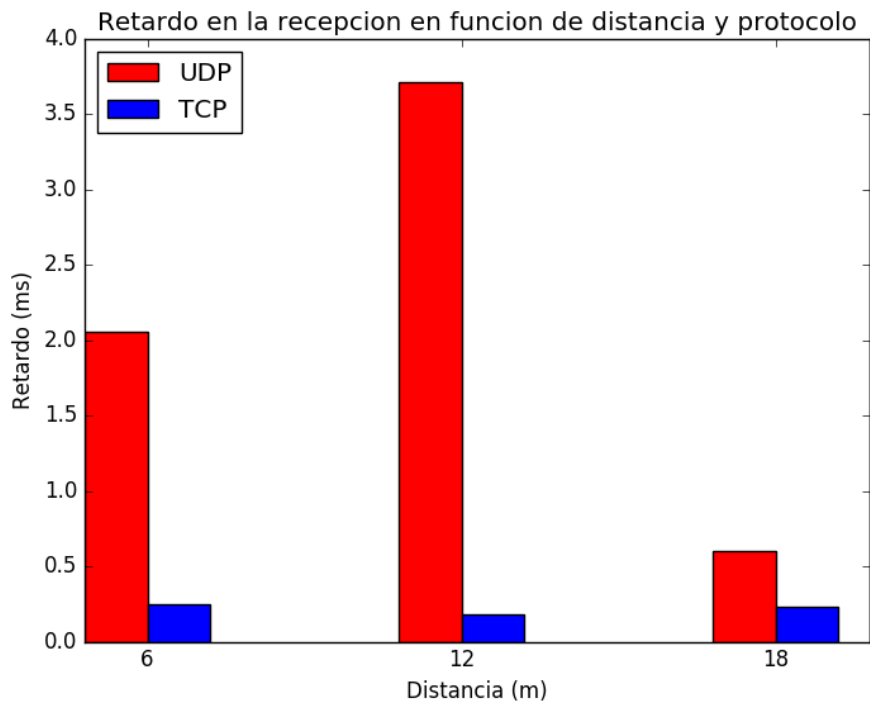


Figura 42. Pruebas E.3 y E.4: Retardo en la recepción de paquetes en función de la distancia y el protocolo de transporte utilizado

### 4.3. Pruebas D.1 y D.2: UDP y TCP con dongle



Figura 43. UAV en vuelo durante las pruebas dinámicas

En este apartado se presentan los resultados obtenidos en las pruebas dinámicas 1 y 2. En ellas, el dongle USB se colocó en la parte posterior del dron según la línea de visión. Dicho dispositivo utiliza el estándar IEEE 802.11n (ver Figura 21). El fichero log correspondiente a la prueba D.1 se muestra en la Figura 44, y refleja la terna de distancias con sus respectivos mensajes (a seis metros, aquellos enviados desde las 17:40:26 hasta las 17:41:28; a doce metros, los que transcurren desde las 17:41:35 y las 17:42:45; a dieciocho metros, aquellos desde las 17:43:00 y las 17:43:56).

Por otro lado, en la Figura 45 se presenta el log correspondiente a la prueba D.2 (transmisiones basadas en el protocolo TCP). Entre las 17:46:58 y las 17:48:00 se mandaron mensajes a seis metros; a doce, entre las 17:48:08 y las 17:49:16; por último, los mensajes a dieciocho metros fueron enviados a partir de las 17:49:25.

```

Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:40:26: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:40:42: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:40:49: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:40:55: walking up the road
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:01: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:07: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:21: They are so happy
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:28: they aren't saying anything, these boys.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:35: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:42: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:49: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:56: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:42:02: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:42:08: walking up the road
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:42:15: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:42:22: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:42:28: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:42:38: They are so happy
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:42:45: they aren't saying anything, these boys.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:00: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:06: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:19: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:25: walking up the road
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:31: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:37: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:42: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:49: They are so happy
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:56: they aren't saying anything, these boys.

```

Figura 44. Fichero log correspondiente a la prueba D.1

```

Esperando conexiones...
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:46:58: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:04: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:13: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:20: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:26: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:32: walking up the road
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:35: to deliver the newspaper
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:42: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:48: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:54: They are so happy
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:00: they aren't saying anything, these boys.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:08: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:19: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:24: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:31: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:38: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:46: walking up the road
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:53: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:58: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:49:05: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:49:10: They are so happy
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:49:16: they aren't saying anything, these boys.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:49:25: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:50:28:
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:50:28:
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:50:28:
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:50:28:

```

Figura 45. Fichero log correspondiente a la prueba D.2

Obsérvese en las siguientes figuras la caída en la tasa de recepción que se experimenta a la distancia máxima (ver Figura 46), y que arrastra por tanto al resto de las métricas dando lugar a un caudal ínfimo. Los retardos, asimismo, fueron considerablemente mayores que los obtenidos en las pruebas de comunicaciones estáticas.

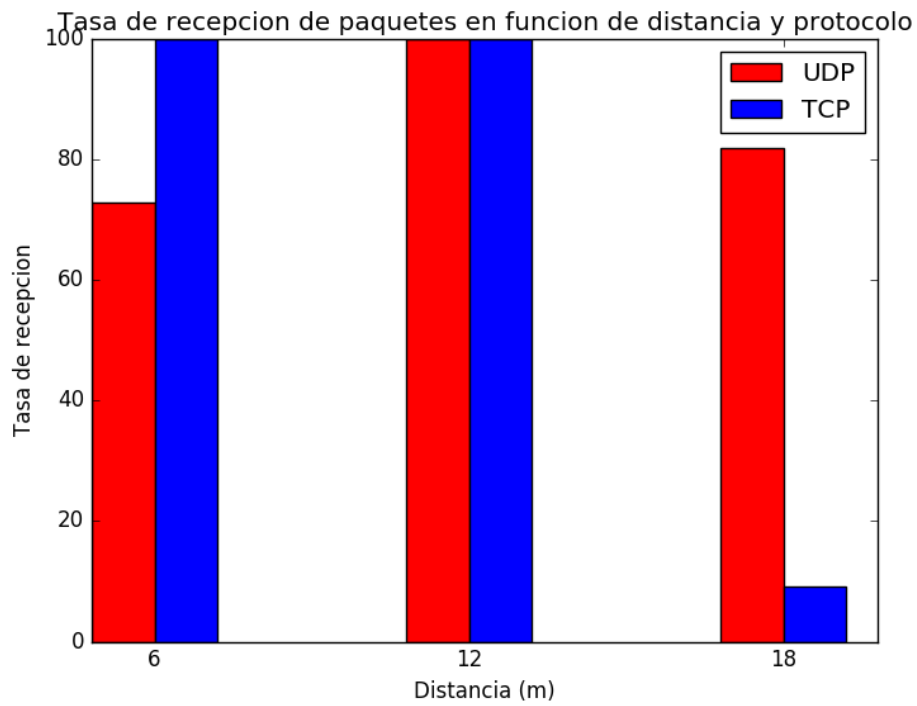


Figura 46. Pruebas D.1 y D.2: Tasa de recepción de paquetes en función de la distancia y el protocolo de transporte utilizado

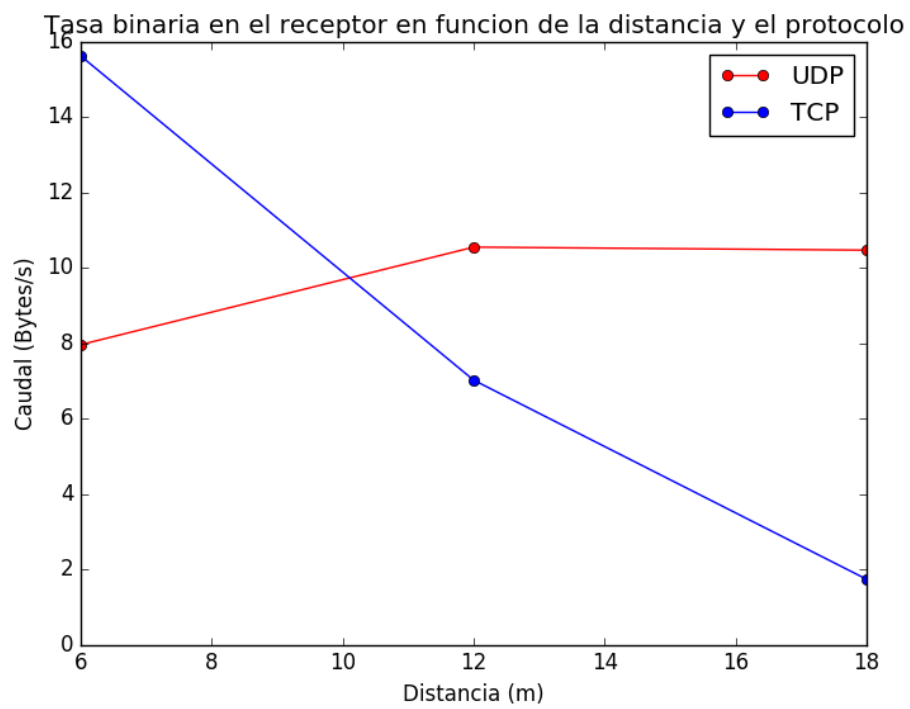


Figura 47. Pruebas D.1 y D.2: Tasa binaria de recepción de paquetes en función de la distancia y el protocolo de transporte utilizado

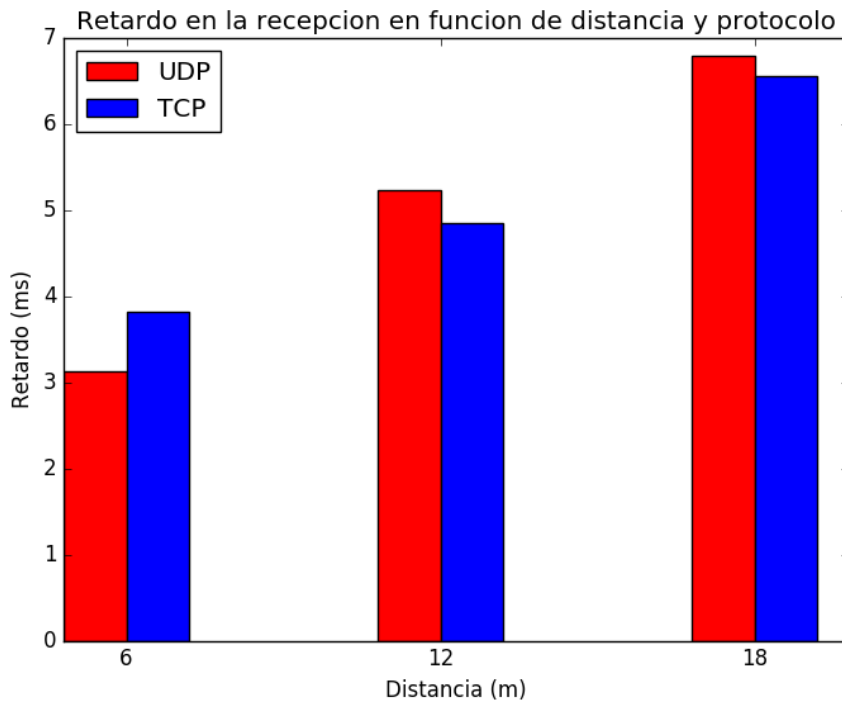


Figura 48. Pruebas D.1 y D.2: Retardo en la recepción de paquetes en función de la distancia y el protocolo de transporte utilizado

#### 4.4. Pruebas D.3 y D.4: UDP y TCP con transmisor integrado

A continuación se presentan los resultados obtenidos de las pruebas dinámicas 3 y 4. El fichero .log correspondiente al protocolo UDP corrobora que todos los mensajes transmitidos desde la GCS han llegado a su destino correctamente (ver Figura 49).

```

Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:40:44: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:40:53: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:00: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:07: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:13: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:21: walking up the road
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:28: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:35: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:44: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:50: They are so happy
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:41:57: they aren't saying anything, these boys.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:42:10: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:42:17: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:42:25: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:42:37: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:42:43: walking up the road
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:42:49: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:42:56: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:02: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:09: They are so happy
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:15: they aren't saying anything, these boys.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:23: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:35: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:41: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:48: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:43:54: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:44:00: walking up the road
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:44:06: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:44:12: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:44:19: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:44:25: They are so happy
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:44:32: they aren't saying anything, these boys.

```

Figura 49. Fichero log correspondiente a la prueba D.3



El fichero muestra las tres ráfagas de mensajes enviadas a distintas distancias (a seis metros, entre las 17:40:44 y las 17:41:57 horas; a doce metros, entre 17:42:10 y 17:43:15; y a dieciocho metros, desde las 17:43:23 y las 17:44:32). Como se refirió en el apartado 3.3.5, la simplicidad del protocolo UDP hace que, a la hora de calcular el número de paquetes transmitidos con éxito, el fichero log sirve de mayor ayuda que la captura de Wireshark.

En lo que corresponde a las transmisiones realizadas con el protocolo TCP, el fichero log se muestra en la Figura 50. Se ve, también, cómo todos los mensajes han llegado correctamente al servidor, independientemente de la distancia a la que se encontraran (nuevamente cinco -entre las 17:46:07 y las 17:47:10;- 10 -entre las 17:47:18 y las 17:48:26 - y 15 metros - entre las 17:48:35 y las 17:49:40 -).

```
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:46:07: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:46:13: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:46:20: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:46:27: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:46:33: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:46:40: walking up the road
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:46:45: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:46:51: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:46:58: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:04: They are so happy
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:10: they aren't saying anything, these boys.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:18: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:25: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:31: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:37: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:44: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:52: walking up the road
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:47:59: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:06: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:13: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:19: They are so happy
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:26: they aren't saying anything, these boys.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:35: So early it's still almost dark out.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:42: I'm near the window with coffee,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:50: and the usual early morning stuff
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:48:56: that passes for thought.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:49:03: When I see the boy and his friend
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:49:10: walking up the road
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:49:16: to deliver the newspaper.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:49:22: They wear caps and sweaters,
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:49:28: and one boy has a bag over his shoulder.
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:49:34: They are so happy
Mensaje recibido de 192.168.1.137 el día 07-09-2017 a las 17:49:40: they aren't saying anything, these boys.
```

Figura 50. Fichero log correspondiente a la prueba D.4

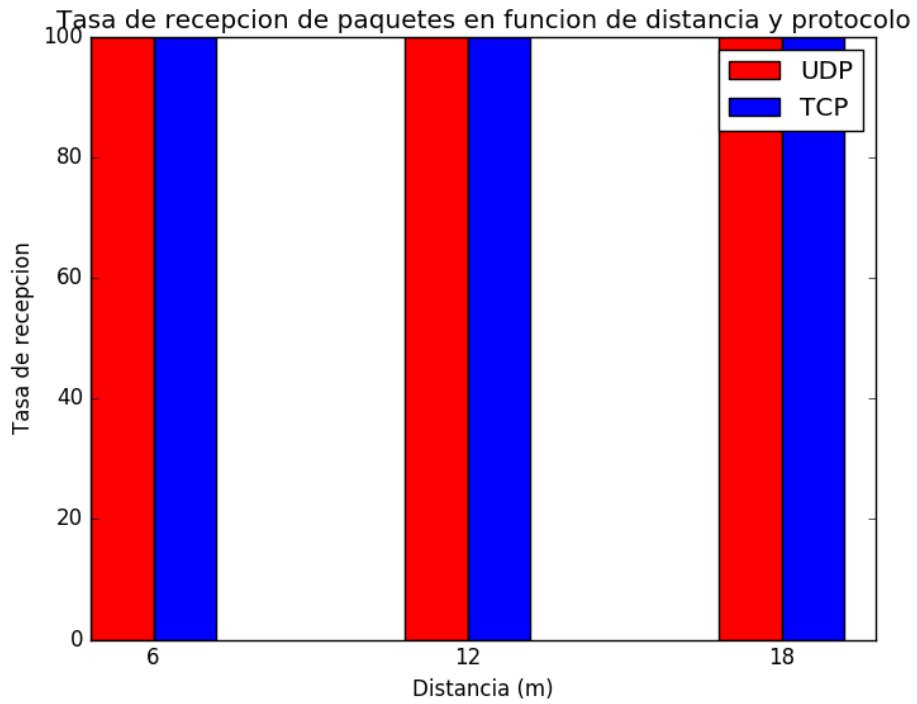


Figura 51. Pruebas D.3 y D.4: Tasa de recepción de paquetes en función de la distancia y el protocolo de transporte utilizado

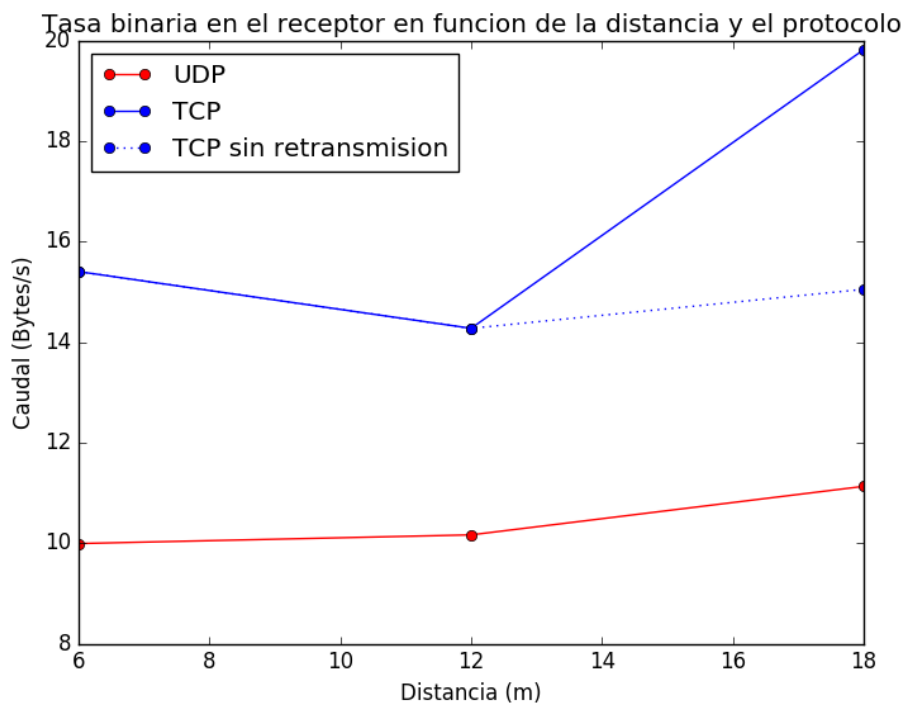


Figura 52. Pruebas D.3 y D.4: Tasa binaria de recepción de paquetes en función de la distancia y el protocolo de transporte utilizado

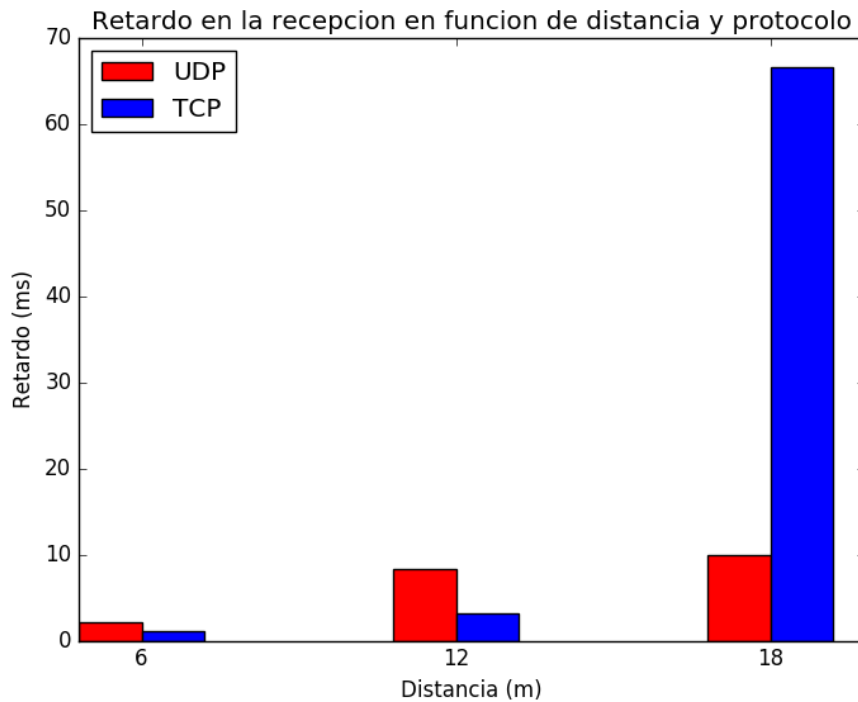


Figura 53. Pruebas D.3 y D.4: Retardo en la recepción de paquetes en función de la distancia y el protocolo de transporte utilizado

La Figura 51 refleja gráficamente lo que los ficheros logs habían atestiguado anteriormente: que todos los mensajes enviados, independientemente del protocolo utilizado, llegan de forma íntegra al UAV.

Obsérvese, de la Figura 52 que si bien la tasa binaria de TCP a una distancia de 18 metros aumenta de manera súbita, esto se debe a dos circunstancias: en primer lugar, la mayor carga de las cabeceras TCP, con muchos más campos que UDP; en segundo; fenómenos de retransmisión que, por la naturaleza de este protocolo, aseguran que los mensajes lleguen a su destino correctamente. Si se realiza una abstracción de estas retransmisiones, la tasa binaria en el receptor se mantiene aproximadamente constante en las tres distancias testadas.

Por último, el retardo (ver Figura 53) sí ha demostrado ser un fenómeno que depende estrechamente de la distancia entre estaciones. Se concluye que a 18 metros el tiempo medio que tardan los mensajes incrementa drásticamente, si bien este fenómeno está íntimamente relacionado con las retransmisiones que se explicaron en el párrafo anterior.

# 5 ANÁLISIS DE RESULTADOS Y FUTUROS TRABAJOS

---

*We know what we are, but know not what we may be*

*- William Shakespeare, dramaturgo -*

**P**ueden extraerse varias conclusiones en relación con los resultados consignados en el apartado anterior. Para una mayor claridad, estas se desglosarán según el escenario en el que se hayan desarrollado: estático o dinámico.

En lo que concierne a las pruebas estáticas, fue el dongle USB externo el que en general proporcionó mejores resultados. Sin embargo, conviene recalcar que en todos los casos se consiguió una tasa de recepción del cien por cien. Tanto el protocolo UDP como TCP rindieron de forma correcta, si bien los retardos experimentados con el uso del primero fueron mayores. Las tasas de transmisión fueron superiores en TCP, pero a efectos de datos útiles, resultaron similares en ambos casos. Puede dirimirse, por tanto, que en casos en los que las RPi se mantengan en posiciones estáticas, o en entornos ambientales resguardados, el módulo Wi-Fi integrado es la opción idónea.

Las pruebas dinámicas produjeron resultados muy distintos a los de las estáticas. Las tasas de recepción de paquetes se redujeron dramáticamente, y en concreto, el uso del dongle USB dio lugar a una caída del ratio cercana a un 90 por ciento a la máxima distancia. Con este dispositivo, incluso distancias menores presentaron pérdidas de paquetes. En condiciones de tasas de recepción altas, las velocidades de transmisión resultaron asimilables a las conseguidas en las pruebas estáticas, así como los retardos. Nuevamente, el módulo Wi-Fi integrado se reveló como la mejor alternativa de entre las posibles.

Y, ¿qué conocimiento puede apprehenderse de esto? Lo cierto es que comunicaciones elásticas basadas en UDP, no demasiado sensibles a retrasos o pérdidas puntuales de paquetes, podrían desenvolverse de forma aceptable. Así, la transmisión de audio o vídeo debieran ser tenidas en cuenta, tratándose además, sobre todo esta última, de una de las aplicaciones imperantes de los drones en la actualidad. Esto refuerza, por tanto, la constancia de que un sistema de comunicaciones basado en Raspberry Pi debe considerarse una opción viable para ser implementada en vehículos aéreos no tripulados, sobre todo si se mejora en lo concerniente a la seguridad y se atiende a las posibles mejoras que debieran desarrollarse en futuras ampliaciones de este proyecto.

A este respecto, se considera importante calibrar los efectos de la interferencia electromagnética sobre la calidad de las comunicaciones. Como se ha referido en este trabajo, el UAV utilizado contiene una gran cantidad de componentes electrónicos que generan espurios (ruido electromagnético). En consecuencia, posibles vías de aplicación encaminadas a asegurar la compatibilidad electromagnética del dron, reforzando aislamientos entre dispositivos, resultarían interesantes. El hecho de utilizar el estándar IEEE 802.11n, que transmiten tramas de forma omnidireccional para conseguir una conectividad más rápida, agrava el problema de la interferencia electromagnética, debido al uso de múltiples radios en un mismo AP [31].

Con respecto a los extremos del sistema de comunicaciones utilizado, es cierto que el módulo integrado que incorpora el modelo 3B de Raspberry Pi presenta un desempeño razonable teniendo en cuenta la gran presencia de interferencias electromagnéticas en el ambiente, siendo en líneas generales mejor que el dongle utilizado. Sin embargo, esto no es óbice para probar el rendimiento con otros dongles disponibles en el mercado (ver Tabla 4), comprobando si las métricas que han centrado este trabajo son mejores en estos últimos.

Asimismo, trabajar con el dron en aplicaciones concretas, tales como videovigilancia, aportaría una visión mucho más fidedigna del rendimiento de las comunicaciones inalámbricas. La posibilidad de integrar en la Raspberry Pi multitud de periféricos, tales como cámaras y una ingente variedad de sensores, dará lugar a la generación de una gran cantidad de datos (vídeo, mediciones ambientales, etcétera) que deben procesarse y transmitirse sobre la red.

Siguiendo estas indicaciones sería razonable considerar distancias menos conservadoras, posibilitando mayor

versatilidad en los movimientos del dron.

# REFERENCIAS

- [1] Barnhart, R., Hottman, S., Marshall, D. and Shappee, E. (2016). *Introduction to Unmanned Aircraft Systems*. 2nd ed. CRC Press LLC.
- [2] Molisch, A. (2013). *Wireless communications*. 1st ed. Hoboken, N.J.: Wiley, pp.25-34.
- [3] Withers, D. (1999). *Radio Spectrum Management: Management of the Spectrum and Regulation of Radio Services*. 2nd ed. Institution of Engineering and Technology.
- [4] Papadimitriou, G., Nicopolitidis, P., Obaidat, M. S., Pomportsis, A. S. (2002). *Wireless networks*. New York: Wiley.
- [5] Kung, H., Lin, C., Lin, T., Tarsa, S. and Vlah, D. (2010). Measuring diversity on a low-altitude UAV in a ground-to-air wireless 802.11 mesh network. *2010 IEEE Globecom Workshops*.
- [6] Frossard, P., de Martin, J. and Civanlar, M. (2008). Media Streaming With Network Diversity. *Proceedings of the IEEE*, 96(1), pp.39-53.
- [7] Akyildiz, I. and Wang, X. (2009). *Wireless mesh networks*. 1st ed. Hoboken, NJ: John Wiley & Sons.
- [8] Yanmaz, E., Kuschnig, R. and Bettstetter, C. (2013). Achieving air-ground communications in 802.11 networks with three-dimensional aerial mobility. *2013 Proceedings IEEE INFOCOM*.
- [9] Guillén Pérez, A., Sánchez Iborra, R., Cano, M., Sánchez Aarnoutse, J. and García Haro, J. (2016). WiFi networks on drones. *2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT)*.
- [10] Alam, S. and Haas, Z. (2006). Coverage and connectivity in three-dimensional networks. *Proceedings of the 12th annual international conference on Mobile computing and networking - MobiCom '06*.
- [11] Nazrul Alam, S. and Haas, Z. (2015). Coverage and connectivity in three-dimensional networks with random node deployment. *Ad Hoc Networks*, 34, pp.157-169.
- [12] Dobkin, D. (2005). *RF engineering for wireless networks*. 1st ed. Amsterdam: Elsevier/Newnes.
- [13] Li, B., Jiang, Y., Sun, J., Cai, L. and Wen, C. (2016). Development and Testing of a Two-UAV Communication Relay System. *Sensors*, 16(10), p.1696.
- [14] Gupta, P. and Saxena, P. (2010). Energy Consumption in Wireless Ad Hoc Network. *2010 3rd International Conference on Emerging Trends in Engineering and Technology*.
- [15] Liang, C., Jong, J., Stark, W. and East, J. (1999). Nonlinear amplifier effects in communications systems. *IEEE Transactions on Microwave Theory and Techniques*, 47(8), pp.1461-1466.
- [16] Pan, Y. and Xiao, Y. (2006). *Ad hoc and sensor networks*. 1st ed. New York: Nova Science.
- [17] Benchoff, B. (2016). *Pi 3 Benchmarks: The Marketing Hype Is True*. [online] Hackaday. Available at: <https://hackaday.com/2016/03/01/pi-3-benchmarks-the-marketing-hype-is-true/> [Accedido el 2 de septiembre de 2017].
- [18] Geier, J. (2002). *Wireless LANs*. 2nd ed. Indianapolis: Sams.
- [19] Fabregas, F. (2016). *Aprender Raspberry Pi con 100 ejercicios prácticos*. 1st ed. Barcelona: Marcombo.
- [20] Lee, J.S. and Wang, Y.M (2013). Experimental Evaluation of ZigBee-Based Wireless Networks in Indoor Environments. *2013 Hindawi Publishing Corporation Journal of Engineering*.
- [21] Rhodes, B. and Goerzen, J. (2011). *Foundations of Python Network Programming*. Berkeley, CA: John Goerzen, Brandon Rhodes.

- [22] Petrovic, M. and Aboelaze, M. (2003). Performance of TCP/UDP under ad hoc IEEE802.11. *10th International Conference on Telecommunications, 2003. ICT 2003*.
- [23] Desongles Corrales, J. (2012). *Técnicos de informática del Servicio Vasco de Salud-Osakidetza*. Alcalá de Guadaíra (Sevilla): MAD.
- [24] Xylomenos, G. and Polyzos, G. (1999). TCP and UDP performance over a wireless LAN. *IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No.99CH36320)*.
- [25] Stangherlin, K., Filho, R., Lautenschlager, W., Guadagnin, V., Balbinot, L., Balbinot, R. and Roesler, V. (2011). One-way delay measurement in wired and wireless mobile full-mesh networks. *2011 IEEE Wireless Communications and Networking Conference*.
- [26] Dipobagio, M. (2009). An overview on ad hoc networks. *Institute of Computer Science (ICS), Freie Universität Berlin*.
- [27] Frodigh, M., Johansson, P., Larsson, P., (2000). Wireless ad hoc networking – The art of networking without network. *Ericsson Review No. 4, 2000*.
- [28] Mills, D. (1991). Internet time synchronization: the network time protocol. *IEEE Transactions on Communications, 39(10)*, pp.1482-1493.
- [29] Vigorra Treviño, S. (2016). *Diseño, montaje y puesta en marcha de vehículos aéreos multirrotores para su uso en redes de comunicaciones aéreas*. Universidad de Sevilla.
- [30] Cao Zhiwen, Chen Dongsheng, Yu Fei, Wang Haijuan and Zeng Zhihui (2013). EMI suppression of UAV power in aeromagnetic survey. *IEEE Electromagnetic Compatibility Magazine, 2(1)*, pp.45-53.
- [31] Deniau, V., Gransart, C., Romero, G., Simon, E. and Farah, J. (2017). IEEE 802.11n Communications in the Presence of Frequency-Sweeping Interference Signals. *IEEE Transactions on Electromagnetic Compatibility, 59(5)*, pp.1625-1633.





# ÍNDICE DE FIGURAS

---

Figura 1. Fenómeno de interferencia intersímbolo	19
Figura 2. Clasificación de amplificadores según su eficiencia y ángulo de conducción	20
Figura 3. Modelo de red en 2D	23
Figura 4. Modelo de red en 3D	23
Figura 5. Sistema de comunicación inalámbrica basado en relevos	24
Figura 6. Componentes de la Raspberry Pi Model 3	25
Figura 7. Comparativa del rendimiento de 2B y 3B de acuerdo con los tests de Linpack y Dhyrstone	27
Figura 8. Máximo consumo de modelos de RPi durante la puesta en marcha del dispositivo	27
Figura 9. Diagrama de pines en Raspberry Pi 3 Model B	28
Figura 10. Configuración de interfaces de las respectivas RPi	29
Figura 11. Script que fuerza el arranque del servidor VNC al encender Raspberry Pi	30
Figura 12. Opción que permite compartir la conexión inalámbrica con la interfaz Ethernet	31
Figura 13. Direccionamiento proporcionado a la interfaz Ethernet	31
Figura 14. Configuración de la interfaz WLAN	32
Figura 15. Visualización de conexiones activas en Raspbian	32
Figura 16. Pruebas de conectividad en la red ad-hoc	33
Figura 17. Jerarquía de servidores NTP	34
Figura 18. Direcciones de los servidores en el fichero /etc/ntp.conf	34
Figura 19. Modificación en /etc/ntp.conf para escuchar mensajes de difusión NTP	34
Figura 20. Resultado de ejecutar ntpq -pn	35
Figura 21. Características del dongle Wi-Pi	36
Figura 22. Script que ejecuta el arranque del servidor	37
Figura 23. Estructura de la cabecera TCP	38
Figura 24. Estructura de la cabecera UDP	39
Figura 25. Filtro de visualización en Wireshark	40
Figura 26. Interfaz de Wireshark	41
Figura 27. Script que realiza la captura del tráfico	41
Figura 28. Relación de interfaces cuyo tráfico puede capturarse con tshark	42
Figura 29. Opción para fijar una referencia temporal en Wireshark	43
Figura 30. Autenticación basada en el protocolo SSH	44
Figura 31. Escenario de las pruebas estáticas	45
Figura 32. Escenario de las pruebas dinámicas	46
Figura 33. Fichero log correspondiente a la prueba E.1	47
Figura 34. Fichero log correspondiente a la prueba E.2	48
Figura 35. Pruebas E.1 y E.2: Tasa de recepción de paquetes en función de la distancia y el protocolo de transporte utilizado	48

Figura 36. Pruebas E.1 y E.2: Tasa binaria de recepción de paquetes en función de la distancia y el protocolo de transporte utilizado	49
Figura 37. Pruebas E.1 y E.2: Retardo en la recepción de paquetes en función de la distancia y el protocolo de transporte utilizado	49
Figura 38. Fichero log correspondiente a la prueba E.3	50
Figura 39. Fichero log correspondiente a la prueba E.4	50
Figura 40. Pruebas E.3 y E.4: Tasa de recepción de paquetes en función de la distancia y el protocolo utilizado	51
Figura 41. Tasa binaria de recepción de paquetes en función de la distancia y el protocolo de transporte utilizado	51
Figura 42. Pruebas E.3 y E.4: Retardo en la recepción de paquetes en función de la distancia y el protocolo de transporte utilizado	52
Figura 43. UAV en vuelo durante las pruebas dinámicas	53
Figura 44. Fichero log correspondiente a la prueba D.1	54
Figura 45. Fichero log correspondiente a la prueba D.2	54
Figura 46. Pruebas D.1 y D.2: Tasa de recepción de paquetes en función de la distancia y el protocolo de transporte utilizado	55
Figura 47. Pruebas D.1 y D.2: Tasa binaria de recepción de paquetes en función de la distancia y el protocolo de transporte utilizado	55
Figura 48. Pruebas D.1 y D.2: Retardo en la recepción de paquetes en función de la distancia y el protocolo de transporte utilizado	56
Figura 49. Fichero log correspondiente a la prueba D.3	56
Figura 50. Fichero log correspondiente a la prueba D.4	57
Figura 51. Pruebas D.3 y D.4: Tasa de recepción de paquetes en función de la distancia y el protocolo de transporte utilizado	58
Figura 52. Pruebas D.3 y D.4: Tasa binaria de recepción de paquetes en función de la distancia y el protocolo de transporte utilizado	58
Figura 53. Pruebas D.3 y D.4: Retardo en la recepción de paquetes en función de la distancia y el protocolo de transporte utilizado	59

# ÍNDICE DE TABLAS

---

Tabla 1. Clasificación de los vehículos aéreos no tripulados	17
Tabla 2. Estándares 802.11	18
Tabla 3. Comparación de prestaciones entre bandas de 2.4 y 5 GHz	18
Tabla 4. Modelos de adaptadores Wi-Fi para Raspberry Pi	22
Tabla 5. Comparativa de modelos 2B y 3B	26
Tabla 6. Nombre y funcionalidad de los pines de la Raspberry Pi 3 Model B	29
Tabla 7. Configuración de red de las Raspberry Pi	30
Tabla 8. Características de la batería Poweradd MP-1311RR Slim2	36



# CÓDIGOS UTILIZADOS

En esta sección se recopilan todos los códigos escritos en lenguaje Python que han sido necesarios para el desarrollo del proyecto.

## clienteudp.py

*Programa que envía mensajes al servidor valiéndose del protocolo UDP*

```
1. # -*- coding: utf-8 -*-
2. """
3. Created on Thu May 04 19:26:53 2017
4.
5. @author: Carlos
6. """
7.
8. import socket
9. #ip target is introduced
10. UDP_IP = raw_input("Target IP >> ")
11. UDP_PORT = 5005
12. #print target IP and port
13. print "UDP target IP:", UDP_IP
14. print "UDP target port:", UDP_PORT
15. while True:
16.     MESSAGE = raw_input("Message >> ")
17.
18.     #print message
19.     print "Received message:", MESSAGE
20.
21.     sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
22.     sock.sendto(MESSAGE, (UDP_IP, UDP_PORT))
23.
24.     #if message is "close", communication ends
25.     if MESSAGE == "close":
26.         break
```

## servidorudp.py

*Programa que recibe mensajes del cliente UDP.*

```
1. # -*- coding: utf-8 -*-
2. """
3. Created on Thu May 04 19:30:16 2017
4.
5. @author: Carlos
6. """
7.
8. import socket
9. from datetime import datetime
10.
11. #UDP_IP = "127.0.0.1"
12. UDP_IP = "192.168.1.138"
13. UDP_PORT = 5005
14.
15. sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # UDP
16. sock.bind((UDP_IP, UDP_PORT))
17.
18. while True:
19.     data, addr = sock.recvfrom(1024) # buffer size is 1024 bytes
20.     if data == "close":
21.         break
22.
23.     #Si se reciben datos nos muestra la IP y el mensaje recibido
```

```

24.     print "Mensaje recibido de " + str(addr[0]) + " el día " + datetime.now().strftime('%d-%m-%Y') + " a las " + datetime.now().strftime('%H:%M:%S') + ": " + data + " "
25.     #print "received message:", data

```

### clienttcp.py

Programa que envía mensajes al servidor valiéndose del protocolo TCP

```

1.  # -*- coding: utf-8 -*-
2.  """
3.  Created on Sat Mar 04 14:33:13 2017
4.
5.  @author: Carlos
6.  """
7.
8.  import socket
9.  #ip target is introduced
10. TCP_IP = raw_input("Target IP >> ")
11. TCP_PORT=9999
12. #Create socket object
13. s = socket.socket()
14. print "TCP target IP:", TCP_IP
15. print "TCP target port:", TCP_PORT
16. s.connect((TCP_IP, TCP_PORT))
17.
18.
19. while True:
20.     message = raw_input("Message >> ")
21.
22.     #message to be sent
23.     s.send(message)
24.     data=s.recv(1024)
25.     print "Received data: " + data
26.
27.     #if "message is close" communication ends
28.     if message == "close":
29.         break
30.
31. #close socket instance
32. s.close()

```

### servidortcp.py

Programa que recibe mensajes del cliente TCP.

```

1.  # -*- coding: utf-8 -*-
2.  """
3.  Created on Sat Mar 04 14:44:34 2017
4.
5.  @author: Carlos
6.  """
7.
8.  #!/usr/bin/env python
9.
10. #importamos el modulo socket
11. import socket
12. #importamos librería de fechas
13. from datetime import datetime
14.
15.
16. s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
17. s.bind(("", 9999))
18. s.listen(1)
19. print "Esperando conexiones..."
20. sc, addr = s.accept()
21.
22.

```

```

23. while True:
24.
25.     recibido = sc.recv(1024)
26.
27.     #Si el mensaje recibido es la palabra close se cierra la aplicacion
28.     if recibido == "close":
29.         break
30.
31.     #Si se reciben datos nos muestra la IP y el mensaje recibido
32.     print "Mensaje recibido de " + str(addr[0]) + " el día " + datetime.now().strftime('%d-%m-%Y') + " a las " + datetime.now().strftime('%H:%M:%S') + ": " + recibido + " "
33.
34.     #Devolvemos el mensaje al cliente
35.     sc.send(recibido)
36. print "Cerrando"
37.
38. #Cerramos la instancia del socket cliente y servidor
39. sc.close()
40. s.close()

```

## functions.py

Funciones auxiliares utilizadas en otros programas

```

1. # -*- coding: utf-8 -*-
2. """
3. Created on Thu May 11 14:52:42 2017
4.
5. @author: Carlos
6. """
7. import csv
8.
9. def sumarLista(lista):
10.     sum=0
11.     for i in range(0,len(lista)):
12.         sum=sum+lista[i]
13.
14.     return sum
15.
16. def csv_to_array(file_csv, column):
17.     with open (file_csv, "rb") as csvfile:
18.         reader = csv.reader(csvfile)
19.         array = []
20.         for row in reader:
21.             array.append(row[column])
22.         return array
23.
24. def common_elements(list1, list2):
25.     return list(set(list1) & set(list2))

```

## receptionrate.py

Programa que calcula la tasa de recepción de paquetes TCP gracias al número de secuencia de cada una de las tramas. En UDP, al no disponer de un campo similar, este cálculo debe realizarse a través de la observación del fichero log.

```

1. # -*- coding: utf-8 -*-
2. """
3. Created on Fri Jul 28 17:49:02 2017
4.
5. @author: Carlos
6. """
7. from __future__ import division
8.
9. com1=[]
10. data = open(r"C:\Users\Carlos\Downloads\tcp5m.csv")
11. for line in data:
12.     cells = line.split(",")

```

```

13.     com1.append( ( cells[1] ) ) #we're interested in first CSV column
14.
15. data.close()
16. print com1
17. com2=[]
18. data2 = open(r"C:\Users\Carlos\Downloads\tcp5mCLI.csv")
19. for line in data2:
20.     cells = line.split(",")
21.     com2.append( ( cells[1] ) ) #we're interested in first CSV column
22.
23. data2.close()
24. print com2
25.
26. lista=list(set(com1).intersection(com2))
27. tasarec= (len(lista)/len(com2))*100
28. print "La tasa de recepción es del " + str(tasarec) + "%"

```

### throughput.py

Programa que calcula la tasa de transmisión de paquetes. Sirve indistintamente tanto para UDP como TCP.

```

1. # -*- coding: utf-8 -*-
2. """
3. Created on Wed May 17 17:42:21 2017
4.
5. @author: Carlos
6. """
7.
8. from __future__ import division
9. length=[]
10. time=[]
11.
12. data = open(r"C:\Users\Carlos\Downloads\Capturas E1 y E2\udpSERV5m.csv")
13. for line in data:
14.     cells = line.split( "," )
15.     length.append( ( cells[7] ) )
16.     time.append((cells[3]))
17.     time_total=time[-1];
18. data.close()
19.
20. data = [float(i) for i in length]
21. data_total=sum(data)
22. throughput=data_total/float(time_total);
23.
24. print "El caudal de recepción es de " + str(throughput) + " bytes por segundo"

```

### delayudp.py

Programa que calcula el retardo de los paquetes UDP. Debido a las limitaciones de este protocolo este programa únicamente funciona cuando la tasa de recepción es del cien por cien.

```

1. # -*- coding: utf-8 -*-
2. """
3. Created on Sat Sep 09 18:11:31 2017
4.
5. @author: Carlos
6. """
7. from __future__ import division
8.
9.
10. from functions import csv_to_array
11.
12. hortotal=0
13. horasserv=csv_to_array(r"C:\Users\Carlos\Downloads\Capturas E1 y E2\udpSERV15m.csv",3)
14. horascli=csv_to_array(r"C:\Users\Carlos\Downloads\Capturas E1 y E2\udpCLI15m.csv",3)
15.
16. for i in range(len(horascli)):
17.     hor = float(horasserv[i]) - float(horascli[i])

```



```

18.     horttotal=horttotal+hor
19.
20. print "El retardo medio es de " + str (horttotal/len(horascli)) + " segundos"

```

## delaytcp.py

Programa que calcula el retardo de los paquetes TCP. Para ello, primero toma los números de secuencia de los paquetes enviados y recibidos, y entre ellos selecciona los comunes (es decir, los que han llegado correctamente). Una vez conseguido esto, toma los tiempos de transmisión y recepción asociados a cada número de secuencia y, para calcular el retraso medio, resta tiempos elemento a elemento y divide entre el número de paquetes recibidos.

```

1. # -*- coding: utf-8 -*-
2. """
3. Created on Wed May 31 20:20:07 2017
4.
5. @author: Carlos
6. """
7. from __future__ import division
8. from functions import common_elements
9. from datetime import datetime
10.
11.
12.
13. com1=[]
14. data = open(r"C:\Users\Carlos\Downloads\Capturas E1 y E2\tcpCLI15m.csv")
15. for line in data:
16.     cells = line.split( "," )
17.     com1.append( ( cells[1] ) ) #we are interested in column with Sequence Number
18.
19. data.close()
20. print com1
21. com2=[]
22. data2 = open(r"C:\Users\Carlos\Downloads\Capturas E1 y E2\tcpSERV15m.csv")
23. for line in data2:
24.     cells = line.split( "," )
25.     com2.append( ( cells[1] ) ) #we are interested in column with Sequence Number
26.
27. data2.close()
28. print com2
29. c=common_elements(com1,com2)
30. print c
31. lines_c=set(c)
32.
33.
34. #calculamos índices asociados a valores de secuencia
35. ind=[]
36. for i in lines_c:
37.
38.     ind.append(com1.index(i))
39.
40. print ind
41. horas1=[]
42. horas2=[]
43.
44. ind2=[]
45. for i in lines_c:
46.
47.     ind2.append(com2.index(i))
48.
49. print ind2
50. horttotal=0
51. with open( r"C:\Users\Carlos\Downloads\Capturas E1 y E2\tcpCLI15m.csv") as csvfile:
52.     lines_set = set(ind)
53.     for line_number, row in enumerate(csvfile):
54.         if line_number in lines_set:
55.             #print line_number, row
56.             test=row
57.             #print test
58.             last=test.split(",")[3]

```

```

59.         #lasth=datetime.strptime(last, '%H:%M:%S.%f')
60.         horas1.append(last)
61.         #print last
62.         lines_set.remove(line_number)
63.         print horas1
64.
65. with open (r"C:\Users\Carlos\Downloads\Capturas E1 y E2\tcpSERV15m.csv") as csvfile:
66.     lines_set = set(ind2)
67.     for line_number, row in enumerate(csvfile):
68.         if line_number in lines_set:
69.             #print line_number, row
70.             test=row
71.             #print test
72.             last=test.split(",")[3]
73.             #lasth=datetime.strptime(last, '%H:%M:%S.%f')
74.             horas2.append(last)
75.             #print last
76.             lines_set.remove(line_number)
77.             print horas2
78.
79. for i in range(len(horas2)):
80.     hor = float(horas1[i]) - float(horas2[i])
81.     hortotal=hortotal+hor
82.
83. print "El retardo medio es " + str(hortotal/len(horas2))

```

### receptionrategraph.py

Programa que representa en un gráfico de barras la tasa de recepción de paquetes para UDP y TCP en función de la distancia.

```

1. from __future__ import division
2. import numpy as np
3. import matplotlib.pyplot as plt
4.
5.
6. dists = 3
7. tasaUDP = (90,100,100) #valores de tasa de recepción con UDP
8. tasaTCP = (100,100,10) #valores de tasa de recepción con TCP
9. index = np.arange(dists)
10. bar_width = 0.2
11. opacity = 1
12.
13. plt.bar(index, tasaUDP, bar_width,
14.         alpha=opacity,
15.         color='r',
16.         label='UDP')
17.
18. plt.bar(index + bar_width, tasaTCP, bar_width,
19.         alpha=opacity,
20.         color='b',
21.         label='TCP')
22.
23. plt.xlabel('Distancia (m)')
24. plt.ylabel('Tasa de recepcion')
25. plt.title('Tasa de recepcion de paquetes en funcion de distancia y protocolo')
26. plt.xticks(index + bar_width, ('6', '12', '18'))
27. plt.legend()
28. plt.show()

```

### throughputgraph.py

Programa que representa en un gráfico de líneas la tasa de transmisión o caudal para UDP y TCP en función de la distancia.

```

1. # -*- coding: utf-8 -*-
2. """
3. Created on Sat Sep 09 13:49:15 2017
4.

```

```

5. @author: Carlos
6. """
7. import matplotlib.pyplot as plt
8.
9. distance = ['6', '12', '18']
10. throughputudp = [10.5257549817, 10.9775463322, 11.1464190788]; #caudal UDP
11. throughputtcp = [15.4739364077, 15.8757218445, 16.3311867125]; #caudal TCP
12.
13. plt.plot(distance, throughputudp, marker='o', color='r', label='UDP')
14. plt.plot(distance, throughputtcp, marker='o', color='b', label='TCP')
15. #plt.plot(distance, throughputtcpSIN_RETR, marker='o', ls='dotted', color='b', label='
    TCP sin retransmision')
16. plt.xlabel('Distancia (m)')
17. plt.ylabel('Caudal (Bytes/s)')
18. plt.title('Tasa binaria en el receptor en funcion de la distancia y el protocolo')
19. plt.legend(loc=2)
20. plt.show()

```

### delaygraph.py

Programa que representa en un gráfico de barras el retardo de transmisión de paquetes para UDP y TCP en función de la distancia.

```

1. # -*- coding: utf-8 -*-
2. """
3. Created on Sat Sep 09 19:08:45 2017
4.
5. @author: Carlos
6. """
7.
8. import numpy as np
9. import matplotlib.pyplot as plt
10.
11.
12. dists = 3
13. delayUDP = (0.7424,0.8749,0.7104) #retardos UDP
14. delayTCP = (0.7834,0.515299999999,1.9664) #retardos TCP
15. index = np.arange(dists)
16. bar_width = 0.2
17. opacity = 1
18.
19. plt.bar(index, delayUDP, bar_width,
20.         alpha=opacity,
21.         color='r',
22.         label='UDP')
23.
24. plt.bar(index + bar_width, delayTCP, bar_width,
25.         alpha=opacity,
26.         color='b',
27.         label='TCP')
28.
29. plt.xlabel('Distancia (m)')
30. plt.ylabel('Retardo (ms)')
31. plt.title('Retardo en la recepcion en funcion de distancia y protocolo')
32. plt.xticks(index + bar_width, ('6', '12', '18'))
33. plt.legend(loc=2)
34. plt.show()

```

# MENSAJE TRANSMITIDO

---

En este apartado se recoge el texto que se ha enviado desde el cliente hasta el servidor.

## **Happiness**

*Raymond Carver*

*So early it's still almost dark out.*

*I'm near the window with coffee,  
and the usual early morning stuff  
that passes for thought.*

*When I see the boy and his friend  
walking up the road  
to deliver the newspaper.*

*They wear caps and sweaters,  
and one boy has a bag over his shoulder.*

*They are so happy  
they aren't saying anything, these boys.*