Escuela Técnica Superior de Ingeniería de Telecomunicación

Universidad Politécnica de Cartagena

Departamento Tecnologías de la Información y las Comunicaciones

Trabajo Fin de Grado

Grado en Ingeniería Telemática

ESTUDIO PRÁCTICO DEL USO DE DRONES EN EL CAMPO DE LAS TELECOMUNICACIONES

AUTORA: ANA MARÍA MONTOYA OSETE

DIRECTORA: MARÍA DOLORES CANO BAÑOS

CODIRECTOR: ANTONIO GUILLÉN PÉREZ

FECHA: OCTUBRE 2019

Autora	Ana María Montoya Osete
E-mail del Autora	anamaria.montoya.osete@gmail.com
Directora	María Dolores Cano Baños
E-mail del Directora	mdolores.cano@upct.es
Codirector	Antonio Guillén Pérez
E-mail del Codirector	antonio.guillen@edu.upct.es
Título del TFE	Estudio práctico del uso de drones en el campo de las telecomunicaciones
Descriptores	FANET, redes AD-HOC, drones

Resumen

El advenimiento de las Flying Ad hoc NETworks (FANET) ha abierto una ventana de oportunidad para crear nuevos servicios de valor añadido. Aunque está claro que estas redes comparten características comunes con sus antecesoras, por ejemplo, con Mobile Ad hoc NETworks (MANET) y con Vehicular Ad hoc NETworks (VANET), son varias las características que hacen únicas a las FANET. Estas características distintivas imponen una serie de pautas que deben considerarse para su implementación exitosa. En particular, el uso de FANET para servicios de telecomunicaciones presenta desafíos exigentes en términos de calidad de servicio (Quality of Service, QoS), eficiencia energética, escalabilidad y adaptabilidad. En particular, para el campo de las telecomunicaciones, drones y FANET podrían usarse para proporcionar cobertura WiFi a través de una red aérea (por ejemplo, en situaciones de emergencia) o para sustituir estaciones base inalámbricas cuando éstas no están operativas, entre otros ejemplos. El objetivo de este TFE es evaluar las prestaciones en términos de QoS y QoE del uso de drones para proporcionar cobertura de red a áreas terrestres

Titulación	Grado en Ingeniería Telemática
Departamento	Tecnología de la Información y la Comunicaciones
Fecha de Presentación	Octubre 2019

Agradecimientos

A mi familia por su apoyo durante el desarrollo del proyecto en momentos de desesperación.

A Juan Carlos Aarnoutse por su implicación a la hora de realizar las pruebas con los drones, buscando siempre la mejor alternativa cuando surgían imprevistos.

Y, a Lola, por dirigir este TFG.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

1.1. Introducción

El mundo está en constante evolución, y las tecnologías avanzan conforme el tiempo pasa; surgen nuevos desafíos y hay que buscar soluciones; lo que lleva a una evolución y desarrollo de todo lo que nos podamos imaginar. Han pasado muchos años desde que se inventó el primer teléfono, la primera computadora, desde que se formó la primera red, y varios años después podemos decir que nuestra vida ha mejorado gracias al avance de la tecnología.

Un ejemplo de este avance puede ser el teléfono, desde el primer prototipo realizado por Antonio Meucci en 1854 [1] donde apenas se podían realizar llamadas, se han producido numerosas mejoras, pasando por la telefonía fija, telefonía móvil, hasta llegar al momento actual con los conocidos Smartphone; teléfonos con muchísimas más funciones que las pensadas en un primer momento, podríamos decir que son utilizados para todo menos para llamar. Parece que lo más importante en estos momentos es estar conectado, poder compartir archivos multimedia; y para ello es necesario formar una red que ofrezca un buen ancho de banda; y esto es cada vez más difícil puesto que con el paso de los años los dispositivos que hacen uso de las redes, de las bandas de 2.4 y 5 GHz han crecido de manera exponencial, lo que supone un gran desafío a la hora de dar cobertura.

Las redes y la forma de interconectar los computadores y dispositivos móviles ha sido una gran área de investigación desde que apareció la primera red de computadoras, ARPANET, creada por el departamento de Defensa de los Estados Unidos en 1969 y compuesta por 4 nodos situados en UCLA, SRI, UCSB Y UTA, para ser utilizada como medio de comunicación. [2]

Actualmente, hay muchos protocolos de acceso al medio, de comunicación de enrutamiento y debido a sus diferentes características ofrecen diferentes prestaciones, pudiendo ser utilizados en momentos diferentes de acuerdo al tipo de red y lo que queramos obtener, por ello es necesario evaluarlos y ponerlos a prueba para que así los usuarios tengan acceso al mejor servicio.

En este caso vamos a centrarnos en las redes WiFi, una comunicación muy extendida y que da soporte a millones de usuarios en todo el mundo; más concretamente nos centraremos en un uno de los modos de funcionamiento de este tipo de redes: el modo ad hoc. Este modo de funcionamiento nos permite conectarnos directamente a otros dispositivos y mantener una comunicación, sin necesidad de dispositivos intermedios como vendría siendo un AP. En una red ad hoc pueden participar todo tipos de dispositivos, teniendo en cuenta el tipo de dispositivos que sean podemos dividir las redes ad hoc en tres grandes grupos: redes MANET, redes VANET (englobadas en las MANET) y redes FANET (englobadas en las FANET); en el capítulo 2, hablaremos de estas redes y sus diferencias.

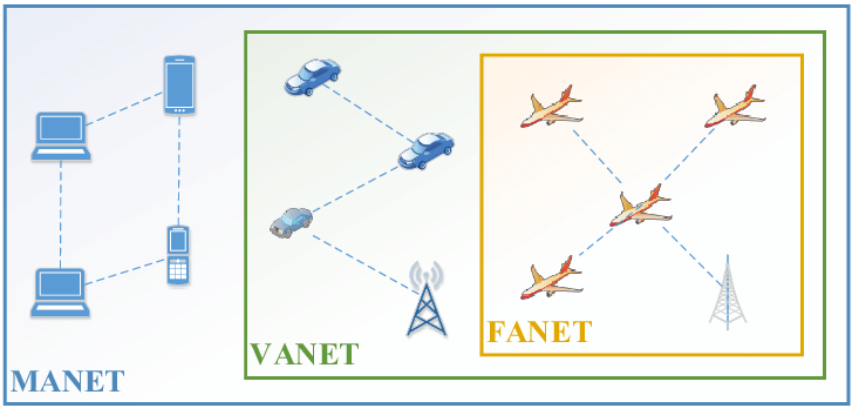

Figura 1. Relación redes MANET, VANET Y FANET.

En este proyecto, y como se específica en el resumen estudiaremos el uso de drones en las comunicaciones ad hoc, es decir, nos centraremos en las redes FANET, que son redes en auge y que permiten un desarrollo e implantación rápida a bajo coste, lo que las hace muy útiles en caso de catástrofe o zonas en las que hay mucha afluencia de gente como un mundial, juegos olímpicos, entre otras cosas.

1.2. Objetivos

El objetivo de este TFG es el estudio y evaluación del uso de los UAV (vehículos aéreos no tripulados, como podría ser un dron) en redes ad hoc, es decir, las redes conocidas como redes FANET. Para ello, analizaremos tres protocolos de encaminamiento:

- OLSR
- B.A.T.M.A.N
- BABEL

En este caso utilizaremos unas WITI Boards, en las que configuraremos los tres protocolos, para más tarde realizar las pruebas necesarias en un campo de vuelo y a partir de los resultados obtenidos, evaluaremos las prestaciones que ofrece cada uno de ellos.

1.3. Fases del proyecto

Para llevar a cabo este trabajo será necesario:

1) Adquisición de conocimientos del estado de la técnica en esta temática.
2) Implementación de algoritmos y métodos de medida en el módulo de comunicación que se añade a los drones.
3) Pruebas experimentales reales en campos de vuelo, con el soporte de pilotos oficiales.
4) Análisis de datos.

1.4. Estructura del proyecto

La memoria de este proyecto está distribuida de la siguiente forma:

- En primer lugar, en el capítulo 1, se realiza una breve introducción, a parte se indica el objetivo y las fases del proyecto.
- Seguidamente, se lleva a cabo un estudio de las redes ad hoc y sobretodo de las redes a tratar en este TFG, las redes FANET.
- En el tercer capítulo, se tratan los tres protocolos a evaluar de forma teórica, es decir, sus características y como trabajan.
- A continuación, en el capítulo 4, se hace un repaso de las tecnologías utilizadas, así como del equipamiento.
- El capítulo 5 contiene la configuración de los protocolos
- En el capítulo 6 hablaremos de las pruebas realizadas y los resultados obtenidos.
- Por último, se encontrarán las conclusiones de los resultados obtenido en el capítulo 7.

REDES AD-HOC

2.1. Redes inalámbricas y modos de funcionamiento

Una red inalámbrica es una red no cableada, como su propio nombre indica donde la información viaja a través del aire hasta llegar a su destino final, pudiendo trabajar en la banda ICM tanto a 2.4 GHz como a 5 GHz. La tecnología usada para este tipo de redes es IEEE 802.11 [3]. El estándar IEEE 802.11 es un conjunto de normas inalámbricas creado por el IEEE (Institute of Electrical and Electronics Engineers) y está formado por varias versiones con diferentes características:

- IEEE 802.11 o 802.11 legacy (1997). Ofrece una velocidad de transmisión de 1, 2, 5.5 y 11 Mbps en la banda de 2.4 GHz. Pudiendo modular en FHSS o DSSS, podemos decir que prácticamente no se usa.
- IEEEE 802.11b (1999). Es una extensión de 802.11 y ofrece 11 Mbps con modulación DSSS.
- IEEE 802.11a (1999). Sigue siendo una extensión de 802.11, con una velocidad de transmisión de 54 Mbps en la banda de 5 GHz y con modulación OFDM. Es incompatible con el estándar 802.11b.
- IEEE 802.11g (2003-2005). Entre 20 y 54 Mbps con las modulaciones DSSS y OFDM, es similar a 802.11a pero en la banda de 2.4 GHz, ofreciendo más alcance y menor consumo que este. Compatible con 802.11b.
- IEEE 802.11n (2007-2009). Entre 450 y 600 Mbps con modulación OFDM; ofrece mejor rendimiento en la banda de 5 GHz. [4]

Las redes inalámbricas surgen a partir de las redes fijas como una necesidad de movimiento, estas redes ofrecen diversas ventajas;

- Permite libertad de movimiento a los usuarios.
- Instalación en zonas difíciles.
- Reducción de tiempos de instalación.
- Reducción de costes ya que no se necesita cableado.
- Características multicast y broadcast.

Pero también presentan sus inconvenientes, ofrecen menor ancho de banda y rendimiento que las redes cableadas. Existen problemas de seguridad, pueden producirse escuchas ilegales y bloqueos intencionados. Ruido, ya que se trabaja en las bandas de uso libre, interferencias, desvanecimientos. Las redes inalámbricas se caracterizan por los siguientes componentes físicos:

- Medio físico de transmisión, en nuestro caso el aire.
- Estaciones (STA), dispositivos con una interfaz wireless, poseen mecanismos de acceso al medio inalámbrico y contacto de radio con el AP.

- Puntos de acceso (AP), se encarga de interconectar a los dispositivos, en una red en modo infraestructura hace de punto intermedio entre dos dispositivos que quiere mantener algún tipo de comunicación.
- Sistema de distribución, permite conectar varios puntos de acceso y así ampliar la cobertura de pueden dar a los usuarios.
- Basic Service Set (BSS), conjunto de estaciones que se comunican unas con otras, la comunicación puede realizarse; bien, directamente (modo ad-hoc, red independiente) o indirectamente (modo infraestructura).
- Extended Services Set (ESS), conjunto de BSS en modo infraestructura unidos por un sistema de distribución.

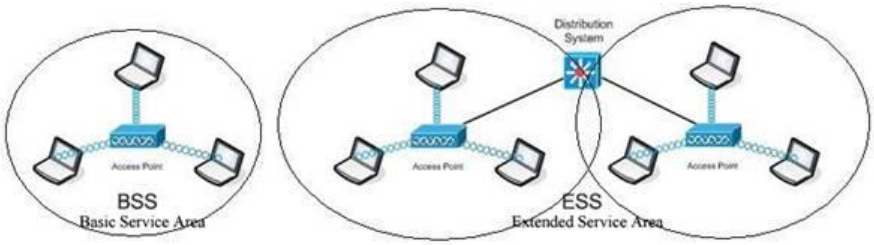

Figura 2. Distribución de una red inalámbrica.

Por tanto y como ya hemos dicho, las redes inalámbricas pueden trabajar en dos modos:

- Modo infraestructura, las redes que trabajan en este modo disponen de un AP a través del cual se realizan todas las comunicaciones entre las estaciones. Si una estación desea comunicarse con otra, la trama viajará hasta el AP, que la retransmitirá a la estación destino.

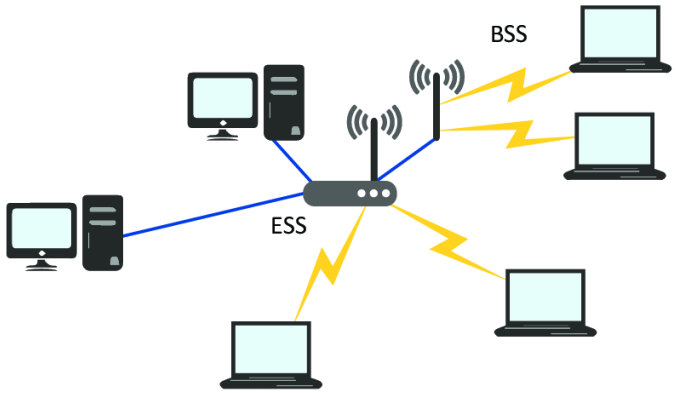

Figura 3. Modo infraestructura.

- Modo ad hoc, el intercambio de tráfico se produce directamente entre las estaciones inalámbricas, no hay puntos de acceso. Las estaciones dentro de una red IBSS se comunican directamente (la red más pequeña está formada por 2 estaciones). [5]

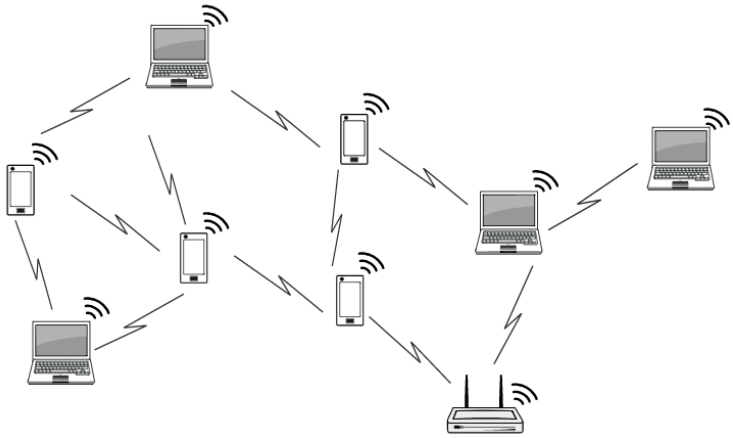

Figura 4. Modo AD HOC.

2.2. Redes ad hoc

Una red AD HOC es aquella que no necesita de una infraestructura existente para sostener la red; en este tipo de redes los dispositivos no deben conectarse a un punto de acceso ni a un enrutador para establecer una conexión inalámbrica con otro dispositivo. Las redes inalámbricas suelen depender de una estación base o un dispositivo WAP para gestionar y dirigir el flujo de datos entre dispositivos inalámbricos, pero en este caso los dispositivos de la red se conectan directamente, por ello también se denominan redes peer-to-peer.

En una configuración ad hoc, la red se construye espontáneamente a medida que los dispositivos se comunican entre sí. Lo ideal es que estos dispositivos se encuentren en un radio de acción muy cercano; sin embargo, la calidad de la conexión y la velocidad de la red se verán afectadas a medida que se añadan más dispositivos a la red.

Debido a que las redes ad hoc requieren una configuración mínima y se pueden implementar rápidamente, tienen sentido cuando se necesita armar una LAN pequeña, generalmente temporal, barata y totalmente inalámbrica. También funcionan bien como un mecanismo de recuperación temporal si falla el equipo para una red en modo infraestructura. [5]

2.2.1. VENTAJAS

- Proporcionan un medio barato de comunicación directa de cliente a cliente, ya que no hay que hacer uso de cables.
- Son fáciles y rápidas de configurar por lo que en caso de emergencias donde se necesita que su instalación sea rápida proporcionan un buen medio de comunicación.
- Ofrecen un buen rendimiento cuando los dispositivos que conforman la red no es demasiado grande. [6]

2.2.2. DESVENTAJAS

- Redes muy susceptibles a ataques, si un atacante se encuentra dentro del rango de señal podrá fácilmente conectarse a la red y dispositivos.
- Al contrario que cuando el número de dispositivos es pequeño, el rendimiento empeora conforme aumentan los dispositivos de la red.
- Los dispositivos no pueden usar Internet a menos que uno de ellos esté conectado a Internet y lo comparta con los demás. Si se habilita el uso compartido de Internet, el cliente que realiza esta función experimentará problemas de rendimiento masivos, especialmente si hay muchos dispositivos interconectados.
- Administrar una red ad-hoc es difícil porque no hay un dispositivo central a través del cual fluya todo el tráfico. Esto empeora conforme el número de dispositivos crece. [6]

2.2.3. TIPOS DE REDES AD HOC

Dentro de las redes ad hoc, podemos encontrar varios tipos. Una red móvil ad-hoc (MANET) es una red móvil de dispositivos inalámbricos autónomos sin red troncal ni infraestructura, pero que presenta características de autoconfiguración. Los MANETs tienen muchas áreas de aplicación, tales como ayuda en caso de desastres, comunicación militar, reuniones de negocios urgentes, etc. La principal ventaja de los MANETs es su portabilidad o movilidad. Las amplias aplicaciones de los MANETs han permitido subcategorías de tecnologías de redes ad-hoc, como las redes ad hoc Vehiculares (VANETs) y las redes ad hoc Voladoras (FANETs). Por lo general, estas redes tienen una alta movilidad con rápidos cambios de topología en comparación con los MANETs típicos, ya que tanto en VANET como en FANET, la mayoría de los nodos son vehículos y UAVs (Unmanned Ariel Vehicles), respectivamente.

Las VANETs son las redes en las que se soporta la comunicación de vehículo a vehículo (V2V) y de vehículo a infraestructura preinstalada. Los principales objetivos de las VANET son la mejora de la eficiencia y la congestión del tráfico, evitar accidentes mediante el acceso a la información.

Flying Ad-hoc Network (FANET) es un tipo especial de MANET con soporte de muy alta movilidad. En FANETs, los nodos son normalmente vehículos aéreos no tripulados (UAVs). [7][8]

2.3. Redes FANET

Una red FANET, es una red ad hoc y más concretamente se clasifica dentro de las redes VANET, es decir, los dispositivos que conforman esa red son vehículos, pero además vuelan, estos dispositivos se denominan UAVs. ¿Y que es un UAV? UAV viene de las siglas Unmanned Ariel Vehicles, es decir, vehículos aéreos no tripulados; también conocidos como RPAS (Remotely Piloted Aircraft System) o comúnmente llamados dron.

Pero, aunque estos tres términos suelen usarse indistintamente, realmente no significan lo mismo, pues hay matices que los diferencian. El término UAV hace referencia a drones de uso militar, por lo tanto, las aeronaves civiles, son las comúnmente denominadas drones; y, por último, un RPAS es el conjunto formado por el dron, los controles que lo gobiernan y la persona que los maneja. Debemos aclarar que un UAV necesariamente no tiene por qué ser controlado por un piloto, por tanto, no siempre un UAV es también un RPA. Aclarado esto, en este proyecto usaremos el término “dron” tanto para facilitar el trabajo como por ser lo más adecuado debido a las pruebas que hemos realizado. [9]

Figura 5. Diferentes tipos de drones.

Un gran desafío en las redes FANET es el enrutamiento ya que debido al rápido movimiento de los UAVs la topología de la red puede cambiar rápidamente. Los protocolos de enrutamiento deben ser capaces de actualizar dinámicamente las tablas de enrutamiento de acuerdo con los cambios de topología. Por tanto, este trabajo consiste en comprobar que protocolo ofrece mejores prestaciones. [10]

PROTOCOLOS

En este capítulo vamos a estudiar un poco la forma de trabajar de los protocolos seleccionados para llevar a cabo este estudio, el motivo por el que vamos a estudiar los protocolos OLSR, B.A.T.M.A.N y BABEL es por la disponibilidad de los paquetes que hay que configurar las WITI Board de las que hablaremos más adelante. Pero antes de ver el funcionamiento de estos protocolos, vamos a ver cómo pueden clasificarse de forma global:

- Protocolos estáticos, tienen tablas de enrutamiento estático y no hay necesidad de actualizar estas tablas.
- Protocolos proactivos, también conocidos como protocolos basados en tablas, actualizan periódicamente las tablas de enrutamiento.
- Los protocolos reactivos, también llamados protocolos bajo demanda, descubren rutas para los mensajes bajo demanda.
- Los protocolos híbridos utilizan protocolos proactivos y reactivos. [11]

3.1. OLSR

El protocolo Optimized Link State Routing o protocolo de enrutamiento de estado de enlace optimizado (OLSR) fue desarrollado como una optimización sobre el protocolo clásico de estado de enlace, adaptándolo para su uso en redes móviles ad hoc. Puesto que es un protocolo proactivo (intercambia información de topología con otros nodos de la red de forma regular) ofrece la ventaja de tener las rutas disponibles en cada momento. Como parte del protocolo cada nodo selecciona un conjunto de sus nodos vecinos como "relés multipunto" (MPR), que serán los responsables de reenviar el tráfico de control, destinado a su difusión en toda la red; lo que ofrece un mecanismo eficiente para el control del tráfico de inundaciones al reducir el número de transmisiones requeridas. OLSR utiliza enrutamiento hop-by-hop, es decir, cada nodo utiliza su información local para enrutar paquetes.

OLSR está diseñado para trabajar de forma totalmente distribuida y no depende de ninguna entidad central. El protocolo no requiere una transmisión fiable de mensajes de control: cada nodo envía mensajes de control periódicamente y, por lo tanto, puede sufrir una pérdida razonable de algunos de ellos. Estas pérdidas se producen con frecuencia en las redes de radio debido a colisiones u otros problemas de transmisión.

3.1.1. Detección de enlaces

Un enlace se describe mediante una interfaz local y una interfaz remota (del nodo vecino); dicha detección se lleva a cabo mediante la emisión periódica de mensajes HELLO a través de las interfaces, formándose así una tabla en cada nodo de enlaces locales.

Los enlaces pueden ser simétricos, es decir, se puede transmitir información en ambas direcciones; o asimétrico, lo que implica que el nodo

escucha los mensajes HELLO de un nodo vecino, pero no se confirma que el nodo vecino pueda recibirlos.

3.1.2. Detección de vecinos

En primer lugar, hay que hacer dos distinciones, existen los nodos de interfaz única y los nodos con múltiples interfaces. En el primer caso, la dirección principal es la dirección de interfaz OLSR, por lo que los nodos deducen directamente de la información de la detección de enlaces el conjunto vecino. Si, por el contrario, el nodo posee múltiples interfaces se necesita información adicional y para ello se utilizan los mensajes de declaración de interfaz múltiple (MID).

En una red con nodos de múltiples interfaces, se necesita información adicional para asignar direcciones de interfaz a direcciones principales (y, por lo tanto, a nodos). Esta información adicional se obtiene a través de los mensajes de declaración de interfaz múltiple (MID). Este mensaje contiene la lista de direcciones de interfaz asociadas a su dirección principal.

3.1.3. Selección MPR

Anteriormente, hemos señalado que los MPR surgen como una optimización de un mecanismo clásico de inundación, para así realizar las inundaciones de los mensajes control en la red de forma controlada, disminuyendo el número de retransmisiones y siendo necesario menor ancho de banda.

¿Cómo elige cada nodo su conjunto MPR? Deben cumplirse dos condiciones, el nodo debe seleccionar su conjunto de MPRs de entre sus nodos vecinos simétricos de 1 salto; y, en segundo lugar, ese conjunto debe poder alcanzar a todos sus vecinos simétricos estrictos de 2 saltos.

3.1.4. Difusión de mensajes de control de topología

Los mensajes de control de topología se difunden con el fin de proporcionar a cada nodo de la red suficiente información sobre el estado del enlace para permitir el cálculo de la ruta.

3.1.5. Tablas de enrutamiento

La tabla de enrutamiento se crea a partir de la información de la tabla de enlaces locales y del conjunto de conjunto de topologías. Cada entrada en la tabla se registra con la siguiente información:

- Dirección del nodo destino
- Siguiente nodo de salto para llegar al nodo destino
- Distancia
- Dirección con la que se puede acceder a uno simétrico

Las tablas de enrutamiento se actualizan cuando se detecta un cambio en alguno de estos conjuntos:

- Conjunto de enlaces
- Conjunto de vecinos
- Conjunto de vecinos de 2 saltos
- Conjunto de topología
- Información de asociación de múltiples interfaces [12]

3.2. B.A.T.M.A.N

El enfoque del algoritmo B.A.T.M.A.N. es dividir el conocimiento sobre las mejores rutas de extremo a extremo entre los nodos de la malla a todos los nodos participantes. Cada nodo percibe y mantiene sólo la información sobre el mejor salto siguiente hacia todos los demás nodos. De esta manera, la necesidad de un conocimiento global sobre los cambios topológicos locales se hace innecesaria. Además, un mecanismo de inundación basado en eventos pero atemporal (atemporal en el sentido de que B.A.T.M.A.N. nunca programa ni da de baja información topológica para optimizar sus decisiones de enrutamiento) evita la acumulación de información topológica contradictoria (la razón habitual de la existencia de bucles de enrutamiento) y limita la cantidad de mensajes topológicos que inundan la malla (evitando así una sobrecarga de tráfico de control). El algoritmo está diseñado para tratar con redes que se basan en enlaces poco fiables.

6.3.1. Algoritmo

En primer lugar, cada nodo transmite mensajes de difusión (mensajes originadores, OGMs) para informar a los nodos vecinos sobre su existencia. Seguidamente, cuando un vecino recibe un OGM lo retransmite para informar a sus vecinos sobre la existencia del iniciador original de este mensaje y así sucesivamente. Por lo tanto, la red está inundada de mensajes del creador. Los OGMs son pequeños (52 bytes incluyendo IP y UDP). Los OGMs contienen al menos la dirección del nodo origen, la dirección del nodo que transmite el paquete, un TTL y un número de secuencia.

Los OGMs que siguen un camino donde la calidad de los enlaces inalámbricos es pobre o saturada sufrirán de pérdida de paquetes o retrasos en su camino a través de la malla. Por lo tanto, los OGM que viajan por buenas rutas se propagarán más rápido y de forma más fiable.

Para saber si un OGM ha sido recibido una o más veces, contiene un número de secuencia, dado por el creador del OGM. Cada nodo retransmite cada OGM recibido como máximo una vez y sólo aquellos que hayan sido recibidos del vecino que ha sido identificado como el mejor siguiente salto (mejor vecino de ranking) hacia el iniciador original del OGM.

De esta manera los OGMs se inundan selectivamente a través de la malla e informan a los nodos receptores sobre la existencia de otros nodos. Un nodo X conocerá la existencia de un nodo Y en la distancia recibiendo sus OGMs, cuando

los OGMs del nodo Y son retransmitidos por sus vecinos de salto único. Si el nodo X tiene más de un vecino, puede decir por el número de mensajes del creador que recibe de forma más rápida y fiable a través de uno de sus vecinos de salto único, qué vecino tiene que elegir para enviar datos al nodo distante.

El algoritmo selecciona entonces a este vecino como el mejor salto siguiente al creador del mensaje y configura su tabla de enrutamiento respectivamente. [13]

3.3. BABEL

Babel es un protocolo de enrutamiento vectorial de distancia, se basa en el protocolo Bellman-Ford aunque incluye mejoras para evitar la formación de bucles. El objetivo del algoritmo de enrutamiento es calcular, para cada fuente S, el árbol de las rutas de la métrica más baja a S.

3.3.1. Detección de vecinos

Los nodos Babel emiten periódicamente mensajes HELLO a todos sus vecinos, además de los mensajes periódicos IHU ("I Heard You") a todos los vecinos de los que ha escuchado recientemente un HELLO (sirven para detectar la bidireccionalidad del enlace); calculando a partir de esta información el coste del enlace con ese nodo vecino. Cada entrada de la tabla de los nodos vecinos contiene la siguiente información:

- Interfaz por la que acceder al nodo vecino.
- Dirección de la interfaz vecina
- Información sobre los paquetes HELLO recibidos recientemente
- Valor del coste de transmisión del último paquete IHU recibido
- Número de secuencia esperado para HELLO.

3.3.2. Selección de ruta

La selección de ruta es el proceso mediante el cual se selecciona una sola ruta para un prefijo determinado que se utilizará para reenviar paquetes y se volverá a anunciar a los vecinos de un nodo. Babel está diseñado para permitir políticas flexibles de selección de rutas. En cuanto a la exactitud del protocolo, la política de selección de rutas debe satisfacer únicamente las siguientes propiedades:

- Una ruta con métrica infinita (una ruta retraída) nunca es seleccionada;
- Nunca se selecciona una ruta inviable.

Puesto que Babel no garantiza naturalmente la estabilidad del enrutamiento, y la configuración de políticas de selección de ruta conflictivas en diferentes enrutadores puede llevar a una oscilación persistente de la ruta. Definir una buena política de selección de rutas para Babel es un problema abierto de investigación.

La selección de la ruta puede tener en cuenta múltiples criterios mutuamente contradictorios; en orden decreciente de importancia, estos son:

- Las rutas con una métrica pequeña deben preferirse a las rutas con una métrica grande;
- Debe evitarse la conmutación de los routers-ids;
- Se deben preferir las rutas a través de vecinos estables en lugar de las rutas a través de los inestables;
- Se deben preferir las rutas estables a las inestables;
- Debe evitarse el cambio de los siguientes saltos.

Después de ejecutar el procedimiento de selección de ruta, se envían las actualizaciones.

3.3.3. Tabla de enrutamiento

Antes de actualizar la tabla se verifica la viabilidad de una actualización lo que garantiza que la ruta no cree ningún bucle de enrutamiento. La condición de viabilidad se aplica a todas las actualizaciones recibidas. La condición de viabilidad compara la métrica de la actualización recibida con las métricas de las actualizaciones enviadas previamente por el nodo receptor; se descartan las actualizaciones con métricas finitas lo suficientemente grandes como para provocar un bucle.[14]

EQUIPAMIENTO Y TECNOLOGÍAS USADAS

Hasta ahora, hemos visto la parte más teórica, ¿pero en que consiste el desarrollo de este proyecto? Para analizar las prestaciones de las redes FANET, vamos a formar una red compuestas por drones, cada uno con su correspondiente WITI, a su vez configuradas con los tres protocolos descritos anteriormente. En este apartado hablaremos del equipamiento y las tecnologías usadas.

4.1. DRONES

Los drones son una parte esencial en este proyecto, en este caso hemos contado con la ayuda y drones del profesor Juan Carlos Aarnoutse y de un piloto externo, Javier. Para volar estos drones hicimos uso del CLUB DE AEROMODELISMO "LOS HALCONES DE LA RAMBLA", situado a las afueras de EL Jimenado, pedanía del municipio de Torre Pacheco; el cual cuenta con una pista asfaltada de unos 100m de longitud.

Figura 6. CDA "Los Halcones de la Rambla"

4.2. WITI BOARD

Los routers que utilizaremos y que anclaremos en cada uno de los drones son las placas de desarrollo WiTi. Esta placa (16cm x 10 cm) fue financiada por una campaña de IndieGoGo que pretendía producir un router abierto y extensible de bajo coste junto con una plataforma NAS. Dicha placa utiliza el router-on-a-chip de alto rendimiento MT7621 y soporta el sistema operativo OpenWrt de código abierto, sistema con el que trabajaremos para configurar los protocolos deseados. Sus características son las siguientes: [15]

- MT7621A Dual-Core 880Mhz (4 hilos)
- 256 MB de memoria (hasta 512 MB)
- 2.4G WiFi (soporta IEEE 802.11b/g/n, hasta 300Mbps)
- 5G WiFi (soporta IEEE 802.11a/n/ac, hasta 867Mbps)
- Puerto WAN x2, puerto LAN x4 (ambos de 1000Mbps)
- SATA 3.0 x2(soporta discos duros de 3.5 pulgadas)
- USB3.0 x1
- microSD x1
- Batería RTC para aplicaciones de almacenamiento

- 30 puertos de expansión (incluyendo USB, I2S, JTAG, UART, GPIO)
- 16MB SPI NOR Flash (Opcional para Flash Nand de mayor tamaño)
- Puerto serie de depuración de 4 pines
- Conector de antena RF x4
- Clave WPS/GPIO x1, Reset Clave x1
- LED de alimentación x1, LED SATA x 1, LED WiFi x 2, LED LAN x4

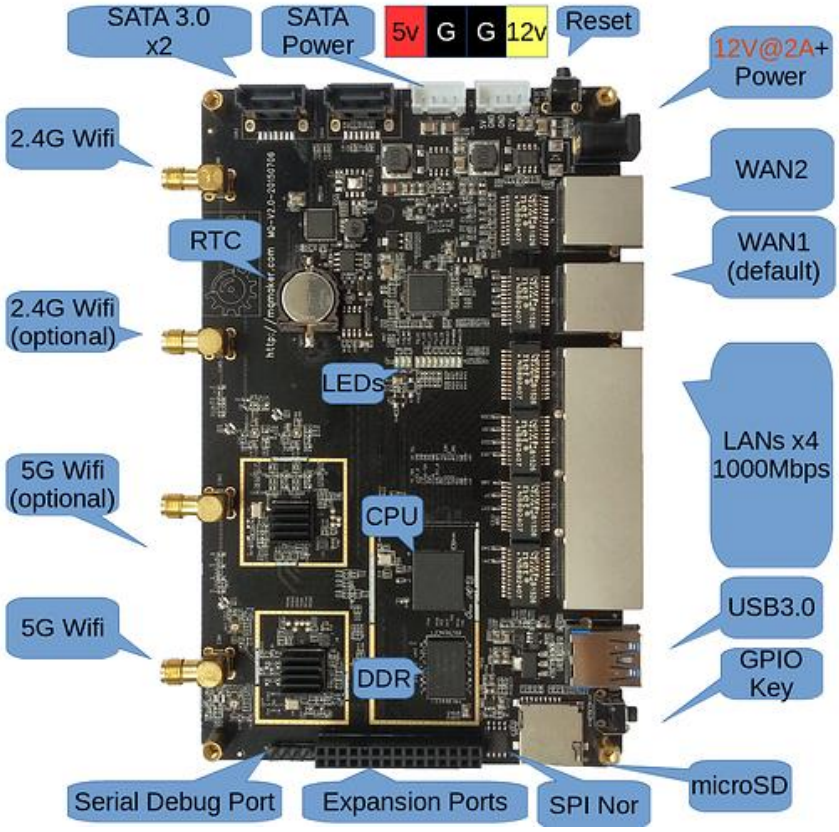

Figura 7. WITI Board

Esta placa tiene muchos propósitos, puede utilizarse como un servidor de almacenamiento instalando el servicio Samba y NFS, gracias a sus 30 puertos de extensión se ofrecen muchas funcionalidades como GPIO (para controlar la luz eléctrica, motores), USB (conectarse a miles de sensores a partir de Arduino), I2S (HiFi inalámbrico). En resumen, WiTi proporciona todas las funciones necesarias, siendo el consumo de electricidad muy bajo. Pero en este caso lo utilizaremos como un router de buen rendimiento. [16]

4.3. OPENWRT

El Proyecto OpenWrt es un sistema operativo Linux dirigido a dispositivos integrados. En lugar de intentar crear un firmware único y estático, OpenWrt proporciona un sistema de archivos totalmente grabable con administración de paquetes. Esto lo libera de la selección y configuración de aplicaciones proporcionadas por el proveedor y permite personalizar el dispositivo mediante el uso de paquetes para adaptarse a cualquier aplicación.

4.1.1. Ventajas de usar OpenWrt

- Extensibilidad: OpenWrt proporciona muchas capacidades que solo se encuentran en dispositivos de gama alta. Sus están estandarizados, por lo que puede replicar fácilmente la misma configuración en cualquier dispositivo compatible.
- Seguridad: los componentes de software de OpenWrt se mantienen actualizados, por lo que las vulnerabilidades se cierran poco después de ser descubiertas.
- Rendimiento y estabilidad: el firmware OpenWrt está hecho de módulos estandarizados utilizados en todos los dispositivos compatibles. Esto significa que es probable que cada módulo reciba más pruebas y corrección de errores que el firmware estándar, que puede ajustarse para cada línea de productos y nunca volver a tocarse.
- Investigación: Muchos equipos usan OpenWrt como plataforma para su investigación sobre el rendimiento de la red. Esto significa que las mejoras de sus experimentos exitosos estarán disponibles primero en OpenWrt, mucho antes de que se incorporen a la línea principal del firmware del proveedor.
- Código abierto / Sin costo adicional: OpenWrt se proporciona sin ningún costo monetario. Ha sido creado en su totalidad por un equipo de voluntarios: desarrolladores y mantenedores, particulares y empresas. [17]

4.4. IPERF3

Una vez formada la red, utilizaremos Iperf para medir el rendimiento de la red entre dos extremos de la comunicación, siendo uno el cliente y el otro el servidor. Debemos tener en cuenta que los flujos de datos UDP nos permiten saber el retardo, paquetes perdidos; mientras que TCP, mide el rendimiento de la carga útil.

4.5. IW

IW es una herramienta de configuración de tarjetas inalámbricas, que sustituye a la herramienta iwconfig. En este caso principalmente la usaremos aparte de para comprobar la configuración de las tarjetas; para configurar la potencia de transmisión y así podremos realizar las pruebas con mayor facilidad. El comando necesario para cambiar la potencia de transmisión y que no emita a máxima potencia es:

```
iw dev <devname> set txpower <auto | fixed | limit> [<tx power in mBm>]
iw phy <phyname> establece txpower <auto | fijo | límite> [<tx power en mBm>]
```

El valor que toma este comando está en mBm en lugar de los dBm comúnmente utilizados. Para pasar de dBm a mBm solo tenemos que multiplicar 100 * potencia en dBm. [18]

CONFIGURACIÓN

Para la instalación de estos protocolos en las WiTi Boards haremos uso del firmware OpenWRT, que ya viene incluido en las WiTi. En algunos casos será necesario tener que actualizar la versión del firmware puesto que para versiones anteriores no hay disponibilidad de paquetes. Para ello se puede trabajar de dos formas, desde LUCI o partir de SSH conectándose al router; en este caso será más sencillo hacerlo desde la interfaz gráfica. Siguiendo los siguientes pasos:

1. Conectar la tarjeta Ethernet de nuestro ordenador con una de las de la WiTi.
2. Ponemos la IP de nuestro router en la barra del navegador, por defecto, estas placas tienen la IP 192.168.200.1. Esto nos llevará a la interfaz gráfica LUCI

   User: root

   Password: root
3. Los pasos anteriores son útiles para acceder la placa, si queremos actualizar la versión del firmware, debemos dirigirnos a la página oficial de OpenWRT (https://openwrt.org/toh/mqmaker/witi) y en el apartado llamado Firmware Download, descargamos la imagen de acuerdo a la versión de nuestra WITI (si es de 256 MB o 512 MB).
4. De vuelta a la interfaz gráfica LUCI, nos dirigimos al apartado firmware y subimos la imagen, después de esto la placa se reiniciará y funcionará con la nueva versión.

Una vez que hemos actualizado la versión de OpenWRT, pasamos a buscar los paquetes de los protocolos bajo estudio, para ello dentro de la misma interfaz nos dirigimos al apartado SOFTWARE,

1. UPDATE PACKAGES (o bien en la ventana de comando > opkg update)
2. Introducimos el nombre del protocolo que queremos buscar, en este caso los paquetes principales a instalar son;
   - olsrd
   - babeld
   - kmod-batman-adv

Seguidamente pasamos a ver la instalación y configuración de dichos protocolos.

5.1. INSTALACIÓN Y CONFIGURACION DE OLSR

Comenzamos instalando los paquetes necesarios; en este caso se ira alternando la ayuda del programa PUTTY, para trabajar a partir de SSH, y el trabajo desde LUCI.

5.1.1. INSTALACIÓN

- opkg update
- opkg install luci-app-olsr luci-app-olsr-services luci-app-olsr-viz olsrd olsrd-mod-arprefresh olsrd-mod-bmf olsrd-mod-dot-draw olsrd-mod-dyn-gw olsrd-mod-dyn-gw-plain olsrd-mod-httpinfo olsrd-mod-mdns olsrd-mod-nameservice olsrd-

mod-p2pd olsrd-mod-pgraph olsrd-mod-secure olsrd-mod-txtinfo olsrd-mod-watchdog olsrd-mod-quagga wireless-tools luci-lib-json kmod-ipip wpad authsae

5.1.2. CONFIGURACIÓN

- cd /etc/config
- vim wireless

```
config wifi-iface
        option device 'radio0'
        option encryption 'none'
        option key 'password'
        option ssid 'OLSR_5G'
        option mode 'adhoc'
        option network 'mesh'

config wifi-iface
        option device 'radio1'
        option encryption 'none'
        option key 'password'
        option ssid 'OLSR_2.4G'
        option mode 'adhoc'
        option network 'mesh'
```

NOTA: Todos los routers deberán de tener la misma configuración, estar en el mismo canal la misma banda y con el mismo SSID para que así puedan verse unos a otros y formar la red.

- vim network

```
config interface 'mesh'
        option proto 'static'
        option ipaddr '192.168.10.X'
        option netmask '255.255.255.0'
```

- Seguidamente vamos a configurar OLSR para que escuche a otros nodos, para ello en primer lugar nos dirigimos a la pestaña SERVICES > OLSR Ipv4 > plugins y marcamos olsrd_jsoninfo.so.1.1 (esto simplemente es para poder administrar desde la interfaz gráfica).
  NOTA: puede que la versión se diferente en otras placas.

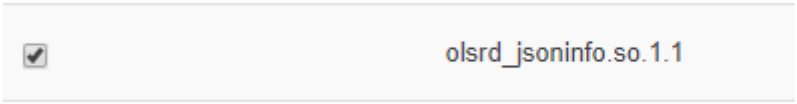

Figura 8. Activación de interfaz gráfica OLSR.

- A continuación, siendo en el apartado de OLSR Ipv4, en la sección Interfaces hacemos click en Edit.

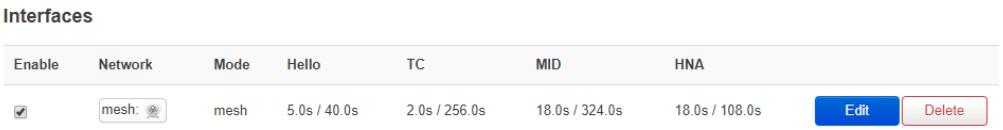

Figura 9. Configuración interfaz OLSR (1)

- Una vez, hecho este debemos marcar mesh (o el nombre elegido) en la opción Network.

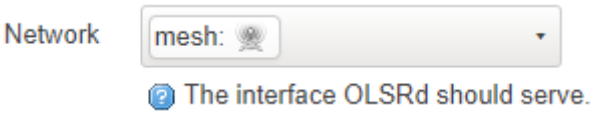

Figura 10. Configuración interfaz OLSR (2)

- Por último tenemos que permitir que pasen los paquetes OLSR a través del firmware, para ello en el menú vamos a NETWORK > FIREWALL y marcamos esta configuración.

Figura 11. Configuración Firewall.

Una vez terminado y si lo hemos configurado bien podremos ir a STATUS > OLSR y veremos los vecinos de nuestra red.

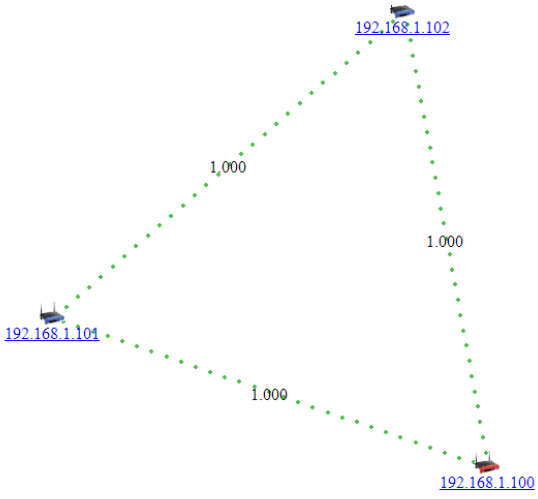

Figura 12. Ejemplo interfaz gráfica OLSR.

5.2. INSTALACIÓN Y CONFIGURACION DE B.A.T.M.A.N

Debemos tener en cuenta la versión que vamos a instalar, ya que la configuración ha ido variando al mismo tiempo que cambiaban las versiones, en nuestro caso, instalamos la última versión disponible para OpenWrt (v. 2018.1-5). Tanto en este caso como con el protocolo babel,

5.2.1. INSTALACIÓN

- opkg update
- opkg install kmod-batman-adv batctl

5.2.2. CONFIGURACIÓN

- cd /etc/config
- vim wireless

```
config wifi-iface
        option device 'radio0'
        option ifname 'adhoc0'
        option ssid 'BATMAN_5G'
        option mode 'adhoc'
        option network 'bat0_hardif_wlan'
        option 'mcast_rate' '18000'

config wifi-iface
        option device 'radio1'
        option ifname 'adhoc1'
        option ssid 'BATMAN_2.4G'
        option mode 'adhoc'
        option network 'bat0_hardif_wlan'
        option 'mcast_rate' '18000'
```

- vim network

```
config interface 'bat0_hardif_wlan'
        option mtu '1532'
        option proto 'batadv'
        option mesh 'bat0'
        option routing_algo 'BATMAN_V'

config interface 'bat0_hardif_eth0'
        option mtu '1532'
        option proto 'batadv'
        option mesh 'bat0'
        option ifname 'eth0'

config interface 'my_bat_vlan1'
        option proto 'batadv_vlan'
        option ifname 'bat0.1'
        option ap_isolation '1'
```

```
config interface 'vlan1111'
        option type 'bridge'
        option stp '1'
        option ifname 'eth1.1111 bat0.1111'
        option proto 'static'
        option ipaddr '192.168.11.X'
        option netmask '255.255.255.0'
        option delegate '0'
```

```
root@OpenWrt:~# batctl n
[B.A.T.M.A.N. adv openwrt-2018.1-5, MainIF/MAC: eth0/ae:e0:10:46:e9:30 (bat0/0a:
3f:10:42:44:82 BATMAN_IV)]
IF             Neighbor              last-seen
      adhoc1     00:00:00:00:00:08    0.640s
```

Figura 13. Uso del comando 'batctl n' para ver los vecinos de una red B.A.T.M.A.N.

5.3. INSTALACIÓN Y CONFIGURACION DE BABEL

5.3.1. INSTALACIÓN

- opkg update
- opkg install babeld

5.3.2. CONFIGURACIÓN

- cd /etc/config
- vim network

```
config interface wlan_24
        option proto 'static'
        option 'ipaddr'   '10.0.0.X'
        option 'netmask'  '255.255.255.0'

config interface wlan_5
        option 'proto'    'static'
        option 'ipaddr'   '10.0.0.X'
        option 'netmask'  '255.255.255.0'
```

- vim wireless

```
config wifi-iface
        option 'device'     'radio0'
        option 'network'    'wlan_5'
        option 'mode'       'adhoc'
        option 'ssid'       'BABEL_5G'
        option 'encryption' 'none'

config wifi-iface
        option 'device'     'radio1'
        option 'network'    'wlan_24'
        option 'mode'       'adhoc'
        option 'ssid'       'BABEL_2.4G'
        option 'encryption' 'none'
```

- vim babeld

```
config general
        option 'random_id'    'true'
        option 'local_port'   '33123'
        list   'import_table' '42'
        list   'import_table' '100'

config interface
        option 'hello_interval' '1'
        option 'update_interval' '16'

config interface
        option 'ifname' 'wlan_24'
        option 'type' 'wireless'

config interface
        option 'ifname' 'wlan5'
        option 'type' 'wireless'

config interface
        option 'ifname' 'mytunnel'
        option 'type' 'tunnel'
        option 'max_rtt_penalty' '100'

config filter
        option 'type'   'redistribute'
        option 'ip'     '0.0.0.0/0'
        option 'eq'     '0'
        option 'proto'  '3'
        option 'action' 'metric 128'

# Accept all routes from neighbours.
config filter
        option 'type'   'in'
        option 'action' 'allow'

# Send all known routes to neighbours.
config filter
        option 'type'   'out'
        option 'action' 'allow'

# Install all routes to the kernel.
config filter
        option 'type'   'install'
        option 'action' 'allow'

# Redistribute all local IP addresses as host routes.
config filter
        option 'type'   'redistribute'
        option 'local'  'true'
        option 'action' 'allow'
```

```
# Don't redistribute any other route from the kernel routing
table.
config filter
        option 'type'   'redistribute'
        option 'local'  'false'
        option 'action' 'deny'

###Redistribuir una ruta dinamica predeterminada
config filter
        option 'type'   'redistribute'
        option 'ip'     '::/0'
        option 'eq'     '0'
        option 'proto'  '3'
        # Only apply this filter when the default route points
   to the eth-main network interface.
        option 'if'     'eth-main'
        # Redistribute with a smallish metric in this case.
        option 'action' 'metric 128'

config filter
        option 'type'   'redistribute'
        option 'ip'     '::/0'
        option 'eq'     '0'
        option 'proto'  '3'
        # Redistribute with a larger metric when the default
route points
        # to the "eth-backup" network interface.
        option 'if'     'eth-backup'
        option 'action' 'metric 512'
```

5.4. RESUMEN IPs.

A continuación, se muestra una tabla con la IP que tiene cada router en cada protocolo.

	OLSR	B.A.T.M.A.N	BABEL	SSID
A (192.168.1.1)	192.168.10.100	192.168.11.10	10.0.0.100	LEDE_5 LEDE_2.4
B (192.168.2.1)	192.168.10.101	192.168.11.11	10.0.0.101	OpenWrt_5G OpenWrt_2.4G
C (192.168.200.1)	192.168.10.102	192.168.11.12	10.0.0.102	OpenWrt5 OpenWrt2.4

Tabla 1. IPs protocolo-router.

PRUEBAS Y RESULTADOS

En primer lugar, vamos a exponer nuestro plan de acción; para la realización de las pruebas debemos tener las placas configuradas y puestas en marcha, para ello tenemos unas baterías y unos adaptadores para poder alimentar la placa con 12V. Puesto que las placas irán sujetas a los drones debemos tener en cuenta que no podemos añadir mucho peso, por tanto, la batería no podía ser de gran tamaño lo cual limitaría mucho el tiempo de vuelo.

6.1. ESCENARIOS

En un principio pensamos en realizar dos escenarios, el primero con solo dos drones y el segundo con tres; el primero es simplemente para comprobar la adaptación de los protocolos, será en el segundo donde podremos estudiar los protocolos de enrutamiento.

6.1.1. Escenario con 2 drones

Este escenario consiste en tener 2 drones volando, cada uno con una placa y sus antenas. El caso es el siguiente, si nuestra idea es probar el protocolo en la banda de 2.4G nosotros nos conectamos al router a través de la banda de 5G para así dejar libre la banda en la queremos hacer las pruebas.

Para obtener una mejor conexión debemos colocar las antenas de una forma específica, es decir, las antenas de la banda que esté a prueba deberán estar hacia arriba o hacia abajo (para la comunicación entre drones) y las otras deberán estar en forma de L, para dar cobertura hacia abajo y que podamos conectarnos para lanzar iperf. [19]

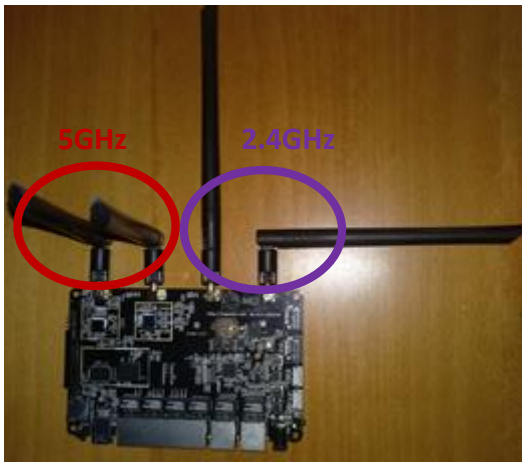

Figura 14. Forma de colocar las antenas para probar un protocolo en la banda de 5GHz.

El objetivo de este escenario es comprobar los paquetes perdidos y el throughput; además de comprobar que protocolo es capaz de reconectar antes. Entonces, para comprobar esto alejaremos y acercaremos los drones desde un punto que se vean hasta otro punto que no se vean; y es aquí donde surge el primer problema a la hora de realizar las pruebas.

El caso es el siguiente, las placas emiten a máxima potencia y una vez que llegamos al campo de vuelo nos dimos cuenta de que para realizar las medidas debíamos alejar tanto los drones que incluso perdíamos visión con ellos. Por tanto, es aquí donde decidimos realizar un estudio de la cobertura (apartado 6.3).

6.1.1.1. Simulación del escenario

Para llevar a cabo este escenario, una vez que estamos en el campo de vuelo y las WITIs están ancladas a los drones (con bridas), los colocamos en el suelo a una distancia de 20m, y conectamos un portátil o móvil a cada extremo. Una vez que estamos conectados, y hemos bajado la potencia de transmisión, entonces los drones elevamos a una altura de 10m; en este momento, las placas tendrán una comunicación total, es entonces cuando dejamos un dron fijo (en la cabecera de la pista) y empezamos a alejar el otro hasta una distancia de 70m (aprox) y 40m, para 2.4 GHz y 5GHz, respectivamente; donde pierden totalmente la comunicación, esperamos unos segundos y volvemos a acercarnos los dos drones hasta que las placas reconectan. Este escenario se repite varias veces para tener una visión más fiable de las perdidas y las prestaciones.

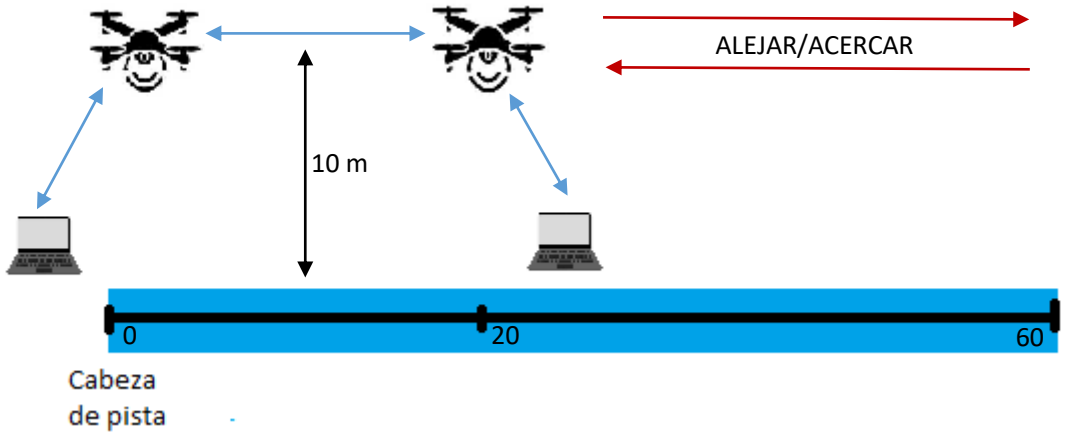

Figura 15. Montaje de la simulación para la prueba de 2 drones.

Antes de empezar a alejar y acercar los drones, debemos lanzar tanto el cliente como el servidor en iperf, para lanzar el servidor nos conectamos a la placa de cabeza de pista y lanzamos el siguiente comando

Iperf3 -s -V --logfile <nombre del fichero donde se guardará>

Mientras que para lanzar el cliente, nos conectamos a la otra placa y lanzamos el siguiente comando

Iperf3 -c <IP router cabecera> -V -t <tiempo en seg> -u <prueba UDP>

6.1.2. Escenario con 3 drones

El segundo escenario consiste en volar tres drones y comprobar la conexión entre ellos siendo uno el intermedio. La prueba será similar a la anterior, volveremos a lanzar iperf en los dos extremos, pero en este caso lo lanzaremos cada 10m y no de forma continuada. En un principio, los dos extremos se comunicarán directamente debido a la cercanía, será una vez pierdan la comunicación con el otro extremo cuando salte el protocolo de enrutamiento y encaminen por el nodo intermedio.

Para comprobar esto, se dispone un script ejecutable que almacena en un archivo la ruta seguida para llegar al punto final de la red.

```
#!/bin/bash

# Crea un archivo con la fecha y guarda la fecha$fecha

if [ $# -ne 2 ]
then
        echo ""
        echo "Tienes que poner 2 argumentos:"
        echo "1. la direccion IP para traceroute"
        echo "2. El numero de iteraciones que quieres."
        echo ""
        exit -1
fi




touch archivoTraceRoute$(date +%Y_%mMES_%dDIA_%H_%M_%S).txt

CONTADOR=0

while [ $CONTADOR -le $2 ]
do
        date >> archivoTraceRoute$(date +%Y_%mMES_%dDIA_%H_%M_%S).txt
        traceroute $1 >>  archivoTraceRoute$(date +%Y_%mMES_%dDIA_%H_%M_%S).txt
        echo "" >> archivoTraceRoute$(date +%Y_%mMES_%dDIA_%H_%M_%S).txt
        ((CONTADOR++))

done
```

Figura 16. Script traceroute.

6.1.2.1. Simulación del escenario

Para llevar a cabo esta simulación, colocamos los dos drones que harán de extremos a una distancia de 10m, volvemos a conectarnos a ello y se vuelvan a una altura de 10m, lanzando pruebas iperf de 60seg cada una de ellas. Una vez terminada la prueba, alejamos los drones otros 10m y así repetidamente hasta que salte el protocolo de enrutamiento. ¿Qué pasa con el tercer router? Este router no irá colocado en un dron, sino que estará a pie de pista a una distancia de 35m esperando a encaminar la información de un extremo a otro. El motivo de que este último no vuele, es la falta de drones y pilotos, ya que por norma un piloto solo puede volar un dron.

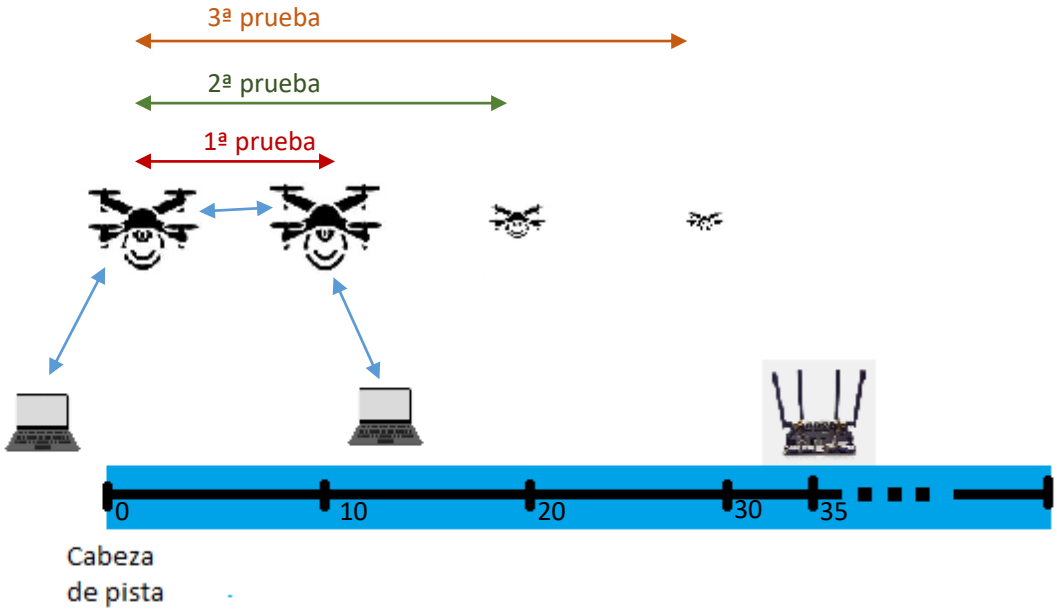

Figura 17. Montaje de la simulación para la prueba de 3 drones (1)

Llegará un momento en el que los quipos no puedan seguir comunicándose directamente porque no se verán, en este caso saltará el protocolo de encaminamiento que estemos probando y se comenzará a enrutar a través del router intermedio situado a pie de pista.

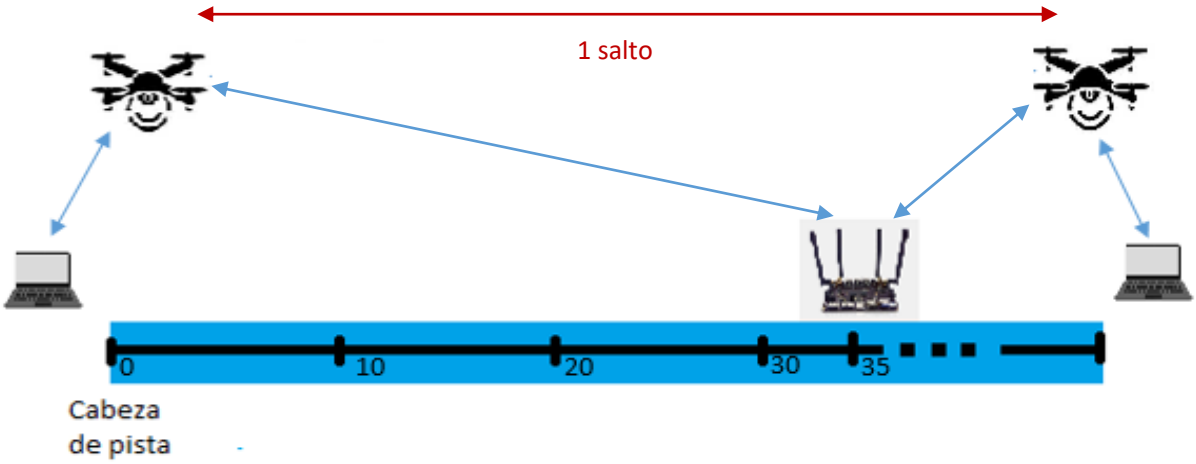

Figura 18. Montaje de la simulación para la prueba de 3 drones (2)

```
traceroute to 192.168.10.102 (192.168.10.102), 30 hops max, 38 byte packets
 1  192.168.10.100 (192.168.10.100)  18.281 ms  2.351 ms  *
 2  192.168.10.102 (192.168.10.102)  1.982 ms  8.854 ms  16.251 ms
```

Figura 19. Comando traceroute (salto intermedio)

6.2. ESTUDIO DE COBERTURA

Como se ha nombrado anteriormente una vez que estábamos en el campo de vuelo vimos la necesidad de realizar un estudio de cobertura antes de continuar con la realización de las pruebas, ya que al estar las placas emitiendo a máxima potencia, teníamos que alejar tanto los drones para que no se viesen que incluso los perdíamos de nuestra visión.

En este caso, el estándar utilizado en la banda de 2.4 GHz es 802.11g y en la banda de 5 GHz es 802.11a. El estándar 802.11a, es un estándar diseñado para la banda de 5 GHz; mientras que el estándar 802.11g es un estándar similar al anterior pero para la banda de 2.4 GHz, ofreciendo este más alcance y menor consumo. Teóricamente hablando el estándar 11g puede llegar a tener un alcance máximo en espacio libre de 460m; y el estándar 11a un alcance de 390m; siendo en condiciones normales de 140 y 120m, respectivamente. [20] [21] [22] [23]

Pero necesitamos, un estudio práctico porque las comunicaciones se ven afectadas por efectos climáticos, ruido, interferencias; y por tanto, lo ideal difiere de la realidad.

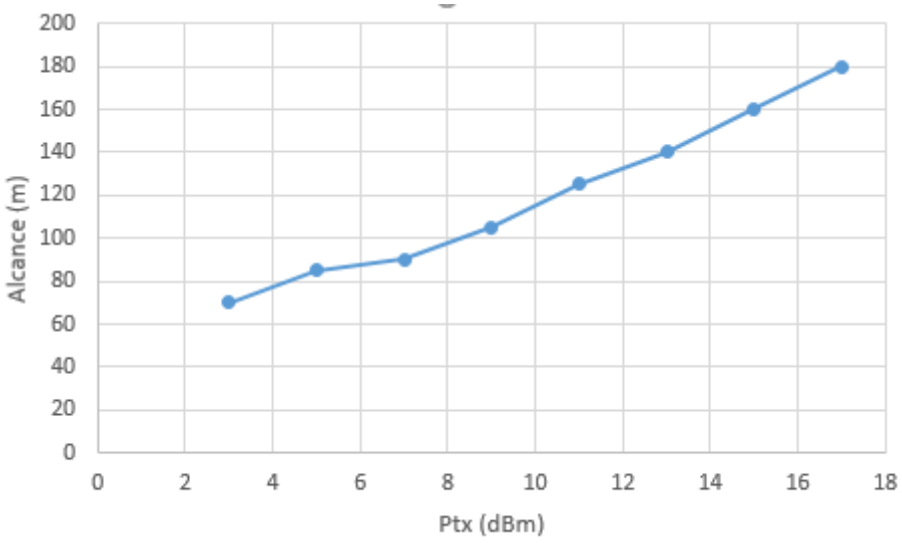

Figura 20. Alcance según la potencia en la banda de 2.4 GHz.

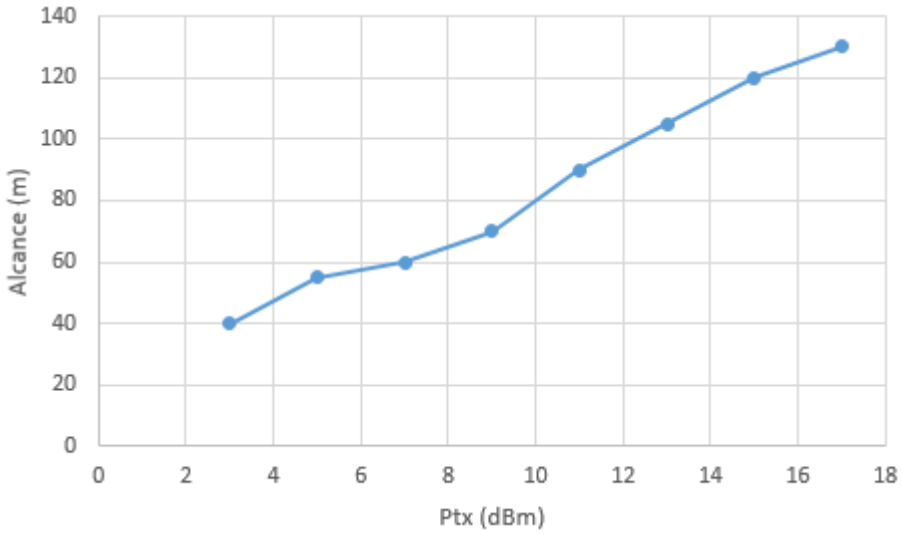

Figura 21. Alcance según la potencia en la banda de 5 GHz.

Hemos decidido usar en ambos casos una potencia de transmisión de 3dB (300 mBm – mBm = dB*100), para facilitar así la realización de las medidas.

6.3. RESULTADOS

A continuación, vamos a analizar los resultados obtenidos de las pruebas del primer escenario.

6.2.1. Packet Loss

La pérdida de paquetes ocurre cuando uno o más paquetes de datos que viajan a través de una red informática no llegan a su destino. La pérdida de paquetes es causada por errores en la transmisión de datos, típicamente a través de redes inalámbricas o por congestión de la red. Se mide como un porcentaje de los paquetes perdidos con respecto a los paquetes enviados. [24]

TCP detecta la pérdida de paquetes y realiza retransmisiones para garantizar mensajes confiables. Podemos ver como al aumentar las retransmisiones baja la ventana (CWD).

```
[ 5] 399.01-400.01 sec   168 KBytes  1.38 Mbits/sec   6   18.4 KBytes
[ 5] 400.01-401.01 sec   218 KBytes  1.78 Mbits/sec   0   25.5 KBytes
[ 5] 401.01-402.01 sec   327 KBytes  2.68 Mbits/sec   0   32.5 KBytes
[ 5] 402.01-403.01 sec   236 KBytes  1.93 Mbits/sec   0   38.2 KBytes
[ 5] 403.01-404.01 sec   215 KBytes  1.76 Mbits/sec   9   32.5 KBytes
[ 5] 404.01-405.01 sec   198 KBytes  1.62 Mbits/sec   5   25.5 KBytes
[ 5] 405.01-406.01 sec   249 KBytes  2.04 Mbits/sec   10   22.6 KBytes
```

En este caso hemos utilizado las pruebas UDP para obtener los paquetes perdidos más fácilmente.

OLSR	2.4 GHz	14 %
	5 GHz	12 %
B.A.T.M.A.N	2.4 GHz	49 %
	5 GHz	54 %
BABEL	2.4 GHz	19 %
	5 GHz	21 %

Tabla 2. Packet Loss en la realización del escenario 1.

Para el escenario 1, podemos ver como BATMAN pierde prácticamente la mitad de los paquetes, mientras que OLSR y BABEL tienen una perdida entre 10 – 20 %, siendo OLSR un poco mejor.

	FRECUENCIAS	10m	20m	30m	40m	50m	60m	70m
OLSR	2.4 GHz	3.3%	0%	0%	27%	11%	19%	
	5 GHz	6.6%	30%	44%	6%	20.4%		
B.A.T.M.A.N	2.4 GHz	0%	0%	13%	22.1%	44%	73%	27%
	5 GHz	14%	84%	97%	21.3%	43%		
BABEL	2.4 GHz	0%	0%	0%	0%	10%	35%	18%
	5 GHz	8%	36%	45.9%	15%	29.7%		

Tabla 3. Packet Loss en la realización del escenario 2.

Para el escenario 2, vemos como B.A.T.M.A.N sigue siendo el protocolo que más paquetes pierde, destacando que OLSR es el protocolo de enrutamiento que antes salta de los tres.

6.2.2. Throughput

El throughput o el rendimiento es la tasa de entrega exitosa de mensajes a través de un canal de comunicación. Los datos a los que pertenecen estos mensajes pueden entregarse a través de un enlace físico o lógico, o pueden pasar a través de un determinado nodo de red. Se suele medir en bits por segundo (bit/s o bps). [24]

	FRECUENCIAS	TX	RX	RENDIMIENTO (%)
OLSR	2.4 GHz	1.05Mbits/sec	916 kbits/sec	87.23 %
	5 GHz		935 Kbits/sec	89.04 %
B.A.T.M.A.N	2.4 GHz		604 kbits/sec	57.52 %
	5 GHz		543 Kbits/sec	51.71 %
BABEL	2.4 GHz		889 Kbits/sec	84.66 %
	5 GHz		824 Kbits/sec	78.47 %

Tabla 4. Rendimiento en la realización del escenario 1.

Para el escenario 1, al igual que en hemos visto en las medidas de paquetes perdidos, vemos como B.A.T.M.A.N no ofrece un rendimiento muy bueno frente a los otros dos protocolos.

	FRECUENCIAS	10m	20m	30m	40m	50m	60m	70m
OLSR	2.4 GHz	97.14%	100%	100%	71.83%	86.75%	80.21%	
	5 GHz	93.14%	69.37%	48.38%	94.28%	78.24%		
B.A.T.M.A.N	2.4 GHz	100%	100%	85.4%	77.91%	48.38%	26.09%	71.3%
	5 GHz	84.66%	15.20%	3.23%	78.06%	55.12%		
BABEL	2.4 GHz	100%	100%	99.68%	100%	88.74%	64.18%	81.9%
	5 GHz	91.38%	63.1%	46.3%	83.4%	68.01%		

Tabla 5. Rendimiento en la realización del escenario 2.

Podemos ver como el rendimiento y la pérdida de paquetes está muy relacionado, cuantos más paquetes se pierden peor es el rendimiento de la red, ya que como hemos dicho el throughput o el rendimiento es la tasa exitosa de entrega de paquetes.

	FRECUENCIAS	10m	20m	30m	40m	50m	60m	70m
OLSR	2.4 GHz	3.3%	0%	0%	27%	11%	19%	
	5 GHz	6.6%	30%	44%	6%	20.4%		
B.A.T.M.A.N	2.4 GHz	0%	0%	13%	22.1%	44%	73%	27%
	5 GHz	14%	84%	97%	21.3%	43%		
BABEL	2.4 GHz	0%	0%	0%	0%	10%	35%	18%
	5 GHz	8%	36%	45.9%	15%	29.7%		

CONCLUSIONES

La conclusión que podemos sacar de los resultados obtenidos es que mientras que OLSR y BABEL no tardan mucho tiempo en volver a tener conexión cuando se alejan demasiado, BATMAN no actualiza sus tablas con la suficiente rapidez lo que hace que se pierdan prácticamente la mitad de los paquetes enviados, bajando así el rendimiento de la red hasta un punto bastante considerable. Por tanto, no sería positivo implantar una red FANET con el protocolo BATMAN si lo que nos interesa es mantener un buen rendimiento.

Por otro lado, vemos como OLSR es el protocolo de enrutamiento más rápido, seguido por BABEL y B.A.T.M.A.N, aunque esto debemos tener en cuenta que se debe a las condiciones de un determinado momento y puede que, en otras condiciones, obtengamos resultados que varían un poco de lo señalado. Vemos B.A.T.M.A.N ofrece peor rendimiento con cambios de topología.

Por otro lado, debemos tener en cuenta que la autonomía de estos routers y de los drones usados no es la suficiente para ciertos escenarios, por lo que habría que buscar drones profesionales con una autonomía mayor y que pudiesen soportar más peso.

En un futuro, sería interesante hacer estas mismas pruebas con más equipos y con otros protocolos, para tener un estudio más completo del trabajo de los drones en las comunicaciones.

BIBLIOGRAFÍA

[1] https://www.history.com/topics/inventions/alexander-graham-bell

[2] ARPANET. [Online]. Available: https://proyectoidis.org/arpanet/.

[3] IEEE 802.11. [Online]. Available: http://www.telecomabc.com/numbers/80211.html

[4] Apuntes asignatura Redes Inalámbricas, Joan García Haro, Departamento Tecnologías de la Información y las Comunicaciones, Universidad Politécnica de Cartagena.

[5] Jochen H. Schiller, "Mobile comunication", 2003. Available: https://doc.lagout.org/Mobile%20Communciations%20by%20Jochen%20Schiller.pdf

[6] https://www.lifewire.com/ad-hoc-mode-in-wireless-networking-816560

[7] Adnan Nadeem, Turki Alghamdi , Ali Yawar , Amir Mehmood, and Muhammad Shoaib Siddiqui; "A Review and Classification of Flying Ad-Hoc Network (FANET) Routing Strategies", 2018. Available: https://pdfs.semanticscholar.org/1143/4c25d5fb8473264b5dfe3a9714004fb5155a.pdf%20%E2%80%A2https://www.universidadviu.es/rpas-uav-drones-cuales-las-diferencias/

[8] https://tools.ietf.org/html/rfc2501

[9] RPAS, UAV y drones: ¿Cuáles son las diferencias? [Online]. Available: https://www.universidadviu.es/rpas-uav-drones-cuales-las-diferencias/

[10] Ozgur Koray Sahingo, "Networking Models in Flying Ad-Hoc Networks(FANETs): Concepts and Challenges", 2013. Available: https://www.researchgate.net/publication/260526688_Networking_Models_in_Flying_Ad-Hoc_Networks_FANETs_Concepts_and_Challenges

[11] https://meshwifi.es/protocolos-de-enrutamiento-en-redes-mesh-malla/

[12] Optimized Link State Routing Protocol (OLSR), rfc3626. [ONLINE]. Available: https://tools.ietf.org/html/rfc3626

[13] OpenMESH. [ONLINE]. Available: https://www.open-mesh.org/projects/open-mesh/wiki/BATMANConcept

[14] The Babel Routing Protocol, rfc6126. [ONLINE]. Available: https://tools.ietf.org/html/rfc6126

[15] WiTi Board. [ONLINE]. Available: https://openwrt.org/toh/mqmaker/witi

[16] https://www.indiegogo.com/projects/witi-board-open-extensible-router-nas-platform#/

[17] OpenWRT. [ONLINE]. Available: https://openwrt.org

[18] https://wireless.wiki.kernel.org/en/users/documentation/iw

[19] Antonio Guillén, "Study of Communications in Unmanned Aerrial Vehicles and Flying Networks", 2017

[20] https://www.geckoandfly.com/10041/wireless-wifi-802-11-abgn-router-range-and-distance-comparison/

[21] https://www.networkworld.es/wifi/80211-estandares-de-wifi-y-velocidades

[22] https://es.ccm.net/contents/789-introduccion-a-wifi-802-11-o-wifi

[23] https://moodle2017-18.ua.es/moodle/pluginfile.php/39723/mod_resource/content/7/conexiones/page_10.htm

[24] https://accedian.com/enterprises/blog/measuring-network-performance-latency-throughput-packet-loss/