Escuela Politécnica Superior de Linares

Grado en Ing. de Tec. de Telecomunicación

UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

DESPLIEGUE DE UNA RED MANET PARA LA CAPTURA DE LA SEÑAL RADIADA POR EMISORES BLUETOOTH LE EN ENTORNOS DE MICROLOCALIZACIÓN

Alumno: **Samuel Ela Nsogo Nsa**

Tutor: Prof. D. Fernando Parra Rodríguez
Depto: Ingeniería de Telecomunicaciones

Junio, 2018

AGRADECIMIENTOS

A la primera persona, que se lo quiero agradecer es a mi tutor Fernando Parra Rodríguez, el difunto que en paz descanse por su valiosa ayuda. En definitiva me ha facilitado todas las herramientas necesarias para completar mi trabajo fin de grado satisfactoriamente.

A mis padres, por haberme proporcionado la mejor educación y lecciones de vida. En especial a mi madre, por haberme enseñado que con esfuerzo, trabajo y constancia todo se consigue, y que en esta vida nadie regala nada.

A mis compañeros de clase, con los que he compartido grandes momentos.

A mis amigos/as por estar siempre a mi lado.

A todos mis familiares, por su apoyo. En especial a mi hermano Bartolomé Abaga Nsogo Nsa, por haberme aportado todo desde pequeño.

También me gustaría agradecer a todas esas personas que ayudan a los demás en los medios sociales, Weblogs y Wikis desinteresadamente.

ABSTRACT

This project is about the search for a solution to process the power levels of the signals received by the iBeacons in a MANET network, and, at the same time, properly and encryptedly transmitting this information to a central information processing system.

The nodes of this network will be based on Raspberry Pi 3 devices with Raspbian operating system which is a variant of Linux Debian. The connection of the nodes will be done with WiFi network technology. . Then we will configure the OLSRD protocol that will be responsible for working in a distributed way to establish the connections between the nodes of the ad-hoc network (MANET), and finally, the basic software that will allow us to develop the bluetooth-based application will be configured in a Raspberry node (in a simple case, receive the burst of data from an iBeacon) and that also allows to send the data received from the Bluetooth devices to a central processing system (which can be in another Raspberry node or it can be a conventional computer). For this we will use the technology called "Websocket" that will be controlled, in addition to the bluetooth communications through the Nodejs development software; which is nothing more than a framework that will allow us to develop the application in Javascript outside of a browser.

By performing the measurements of the MANET network, the system stores the detailed information of the devices, the power levels, as well as other relevant data; with them the application performs a preprocessing with a diagnosis of the network situation and provides recommendations to improve it.

ÍNDICE

1 RESUMEN

Este proyecto se plantea como la búsqueda de una solución para procesar niveles de potencia de las señales recibidas por los iBeacons en una red MANET y, a su vez, transmitir adecuadamente y de forma cifrada esta información a un sistema central de procesado de información.

Los nodos de dicha red estarán basados en equipos Raspberry Pi 3 con sistema operativo Raspbian que es una variante del Linux Debian. La conexión de los nodos será utilizando la tecnología de la red WiFi. A continuación se va configurar el protocolo OLSRD que será encargado de trabajar en forma distribuida para establecer las conexiones entre los nodos de la red ad-hoc (MANET), y por último se va a configurar en un nodo Raspberry el software base que nos permitirá desarrollar la aplicación basada en Bluetooth (en un caso sencillo, recibir la ráfaga de los datos procedentes de un iBeacon) y que además permita enviar los datos recibidos de los dispositivos Bluetooth a un sistema de proceso central (que puede estar en otro nodo Raspberry o bien puede ser un ordenador convencional). Para ello utilizaremos la tecnología denominada “Websocket” que será controlada, además de las comunicaciones Bluetooth mediante el software de desarrollo Nodejs; que no es más que un framework que nos permitirá desarrollar la aplicación en Javascript fuera de un navegador.

Mediante la realización de las mediciones de la red MANET, el sistema almacena la información detallada de los dispositivos, los niveles de potencia, así como otros datos relevantes, con ellos la aplicación realiza un preprocesamiento con un diagnóstico de situación de la red y proporciona recomendaciones para mejorarla.

2 INTRODUCCIÓN

En este capítulo aparece descrita la motivación del proyecto, los objetivos fijados y algunos conceptos esenciales necesarios para la posterior comprensión del resto de los apartados y, finalmente, un desglose de las secciones en que se encuentra estructurado el presente documento.

Como conceptos esenciales se ha considerado relevantes exponer las características detalladas de las redes WiFi como la base fundamental y, que además nos permite poner en marcha este proyecto fin de grado para poder completar la enseñanza del grado en Ingeniería de Tecnología de Telecomunicación.

2.1 Motivación

La localización en interiores es una de las áreas más prometedoras en el campo de la computación móvil. Estos sistemas permiten desarrollar innumerables aplicaciones gracias al posicionamiento de objetos o personas en tiempo real. El número de ellas que actualmente requieren de un sistema de localización en interiores de un edificio es elevado, algunos ejemplos son los siguientes:

- Ámbito sanitario: control de localización en cada momento de pacientes y personal sanitario para la disminución del tiempo de respuesta ante emergencias.
- Ocio y entrenamiento: conocimiento de la localización de visitantes en emplazamientos tales como museos, exposiciones, recintos, deportivos, para ofrecer un mejor servicio y dotarlos de seguridad en caso de extravió de personas.
- Seguridad: para definir zonas dentro de los recintos de forma que solo algunos usuarios autorizados puedan tener acceso a ellas.

Es muy probable que sigan surgiendo nuevas aplicaciones, que, sumadas a las anteriores, necesiten apoyarse en una tecnología apta para realizar localización de objetos en interiores. Sin embargo, actualmente ninguna de las soluciones propuestas ha conseguido el éxito que han alcanzado los sistemas de localización análogos para exteriores, como GPS.

Se ha tratado durante los últimos años de buscar buenas alternativas para tener un buen servicio de localización en el interior de un edificio. El objetivo es satisfacer la

demanda de clientes con necesidad de desarrollar aplicaciones funcionales en esta área de la localización.

Existen múltiples vías tecnológicas de localización en interiores y algunas de ellas se encuentran implantadas y funcionando correctamente. Pero entre todas ellas la tecnología WiFi parece la mejor posicionada para ocupar el hueco existente en sistemas de este tipo.

Ese posicionamiento se debe a que los sistemas de localización sobre redes WiFi suavizan el impacto económico, pues permiten reutilizar estas infraestructuras, que han sido desplegadas previamente con otros fines. En el año 2009, por ejemplo, se duplicó el número de puntos de acceso instalados en los países de la unión europea.

La motivación principal de este proyecto consiste en desarrollar un sistema de microlocalización y seguimiento en interiores con sensores Bluetooth LE sobre una red MANET. El objetivo es aprovechar el despliegue de la red WiFi de fácil acceso para nosotros, para procesar niveles de potencia de las señales recibidas por los iBeacons en una red MANET y, a su vez, transmitir adecuadamente y de forma cifrada esta información a un sistema central de procesado de información.

Mediante la realización de las mediciones de la red MANET, el sistema almacena la información detallada de los dispositivos, los niveles de potencia, así como otros datos relevantes, con ellos la aplicación realiza un preprocesamiento con un diagnóstico de situación de la red y proporciona recomendaciones para mejorarla.

2.2 Conceptos esenciales

En este punto se explica de forma detallada las características de las redes WiFi como la base fundamental y, que además nos permite afrontar de una forma más sencilla este proyecto fin de grado.

2.2.1 Introducción a las redes WiFi

WiFi es una marca comercial de WiFi Alliance. La tecnología WiFi se utiliza con productos certificados que pertenecen a los dispositivos WLAN basados en los estándares IEEE 802.11. En sus inicios WiFi fue pensado para conectar redes locales inalámbricas; sin embargo, actualmente se utiliza para el acceso a internet.

WiFi es una tecnología de área local que alcanza grandes tasas de transmisión según el estándar utilizado, comúnmente opera en la banda de 2.4GHz (banda no licenciada) y opera con modulaciones PSK, QPSK y OFDM. Es una plataforma bastante escalable y de fácil instalación.

2.2.2 WLAN (IEEE 802.11)

Una WLAN es un sistema de comunicaciones de datos que transmite y recibe datos utilizando ondas electromagnéticas, en lugar del par trenzado, coaxial o fibra óptica utilizados en las LAN convencionales, proporcionando conectividad inalámbrica dentro de un área de cobertura.

Con velocidades de transmisión de 11, 54 o 108 Mbps y un rango de entre 50-100m, IEEE 802.11 es actualmente el dominante estándar de redes locales inalámbricas.

Los distintos estándares 802.11 evolucionaron con los años, como se puede ver en la siguiente imagen.

Figura 1: Familia de estándares de las redes WiFi.

- **IEEE 802.11a:** opera en la banda de frecuencia de 5 GHz y proporciona velocidades de hasta 54 Mb/s. Posee un área de cobertura menor y es menos efectivo al penetrar estructuras edilicias ya que opera en frecuencias superiores. Los dispositivos que funcionan conforme a este estándar no son interoperables con los estándares 802.11b y 802.11g que se describen a continuación.
- **IEEE 802.11b:** opera en la banda de frecuencia de 2,4 GHz y proporciona velocidades de hasta 11 Mb/s. Los dispositivos que implementan este estándar tienen un mayor alcance y pueden penetrar mejor las estructuras edilicias que los dispositivos basados en 802.11a.
- **IEEE 802.11g:** opera en la banda de frecuencia de 2,4 GHz y proporciona velocidades de hasta 54 Mbps. Por lo tanto, los dispositivos que implementan este estándar operan en la misma radiofrecuencia y tienen un alcance de hasta 802.11b pero con un ancho de banda de 802.11a.
- **IEEE 802.11n:** opera en la banda de frecuencia de 2,4 GHz y 5 GHz. Las velocidades de datos típicas esperadas van de 150 Mb/s a 600 Mb/s, con un alcance de hasta 70 m. Es compatible con dispositivos 802.11a, b y g anteriores.
- **IEEE 802.11ac:** opera en la banda de 5 GHz y proporciona velocidades de datos que van de 450 Mb/s a 1,3 Gb/s (1300 Mb/s); es compatible con dispositivos 802.11a/n.
- **IEEE 802.11ad:** también conocido como “WiGig”. Utiliza una solución de WiFi de triple banda con 2,4 GHz, 5 GHz y 60 GHz, y ofrece velocidades teóricas de hasta 7 Gb/s.

- **IEEE 802.11ax:** es un tipo de WLAN en el conjunto de IEEE 802.11 de tipos de WLAN. Está diseñado para mejorar la eficiencia espectral general, especialmente en escenarios de implementación densa. Todavía se encuentra en una etapa muy temprana de desarrollo, pero se prevé que tendrá una velocidad máxima de alrededor de 10 Gb/s. IEEE 802.11ax está diseñado para operar en los espectros ya existentes de 2.4 GHz y 5 GHz. Además de utilizar MIMO y MU-MIMO, la nueva modificación introduce OFDMA para mejorar la eficiencia espectral global, y 124-QAM de orden superior soporte de modulación para un mayor rendimiento. Aunque la tasa de datos nominales es solo un 37% más alta que IEEE 802.11ac, se espera que la nueva enmienda logre un aumento de 4 veces el rendimiento del usuario, debido a una utilización del espectro más eficiente. IEEE 802.11ax se lanzará públicamente en algún momento en 2019.

En la siguiente ilustración, se destacan algunas de estas diferencias, en términos de velocidad máxima, frecuencia y compatibilidad con modos aleatorios

Estándar	Velocidad máxima	Frecuencia	Compatible con modelos anteriores
802.11a	54Mb/s	5 GHz	No
802.11b	11 Mb/s	2,4 GHz	No
802.11g	54Mb/s	2,4 GHz	802.11b
802.11n	600 Mb/s	2,4GHz o 5GHz	802.11a/b/g
802.11ac	1,3 Gb/s (1300 Mb/s)	2,4GHz y 5GHz	802.11a/n
802.11ad	7 Gb/s (7000 Mb/s)	2,4GHz, 5GHz y 60 GHz	802.11a/b/g/n/ac

Figura 2: Comparativa de los distintos estándares 802.11

2.2.3 Bluetooth (IEEE 802.15)

Bluetooth opera en la banda ISM a 2.4 GHz. En comparación con WLAN, la velocidad de transmisión es menor (1 Mbps), y el rango es más corto (10-15 m). Por otro lado, es un estándar muy "ligero", con una alta ubicuidad (integrado en la mayoría de los teléfonos, asistentes digitales personales (PDAs), etc. y es compatible con varios servicios de red, además de otros de IP. Las etiquetas de Bluetooth son transceptores de

pequeño tamaño. Como cualquier otro dispositivo, cada etiqueta tiene un identificador único, y éste puede ser utilizado para la localización de la etiqueta.

Figura 3: Estándar IEEE 802.15

2.2.4 Ancho de banda

Este término normalmente es usado para decir la cantidad de datos que se puede enviar o transmitir en la conexión a internet, por ejemplo “mi ancho de banda es de 4Mbps de internet” este término está mal empleado, para referirse a este se debe usar tasa de transferencia de datos.

Pero este ancho de banda no es más que una medida de rango de frecuencia, por ejemplo, si en un dispositivo está usando un rango de 2458 MHz a 2480 MHz, su ancho de banda es de 0.022 GHz o 22 MHz, es la diferencia entre el máximo y menor valor del rango que se está usando. Si el ancho de banda es más grande se pueden enviar más datos en un momento determinado.

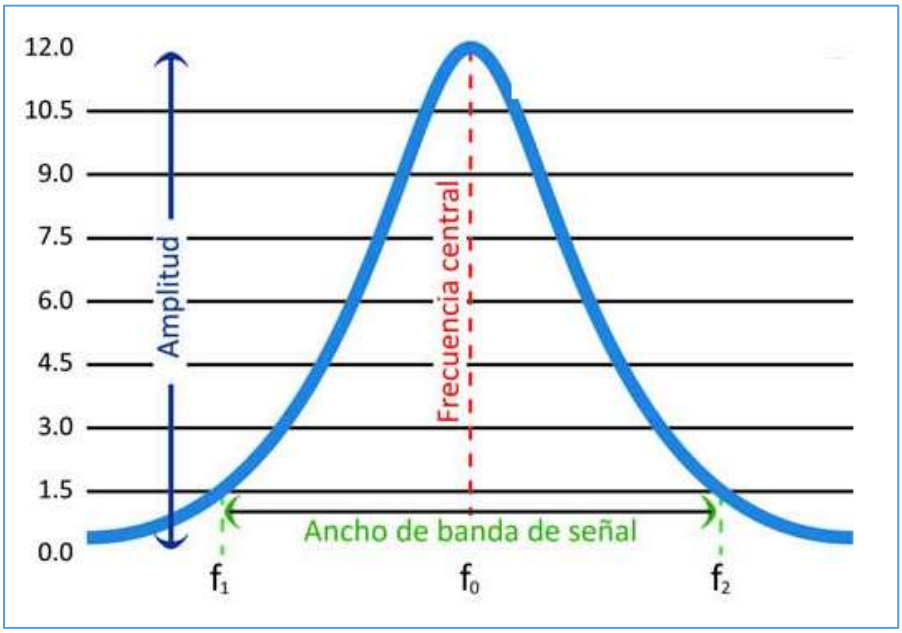


Figura 4: Ancho de banda de la señal.

2.2.5 Frecuencia y canales

Los protocolos 802.11b y 802.11g utilizan la banda 2.4GHz, la cual está dividida en 14 canales iguales. Estos canales tienen un ancho de banda de 22MHz además la distancia entre el principio de un canal y su sucesor es de 5MHz esto lo podemos apreciar en la figura 5; por el cual se determina que si se usa el canal 1 esto se traslapa con su sucesor el canal 2, además de traslaparse con el canal 3 y 4, lo que ocurre es que cuando dos dispositivos que están cerca utilizan canales que se traslapan es que se crean interferencia el uno con el otro y hace difícil la comunicación inalámbrica. A partir de lo anterior y como se muestra en la imagen para la banda 2.4GHz los canales que no se traslapan son el 1,6 y 11 lo que los convierte en los canales ideales para ser utilizados entre nodos cercanos y que no se cree interferencia entre ellos.

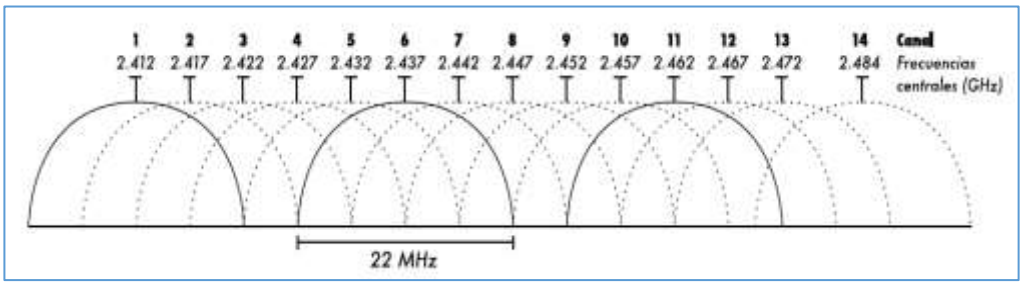


Figura 5: Representación gráfica de la superposición de canales en la banda 2.4GHz.

A pesar de ser 14 canales no los podemos utilizar todos, como ya se dijo anteriormente crean interferencias si se usan los canales sucesivos. Pero también hay un nuevo motivo por el cual no se pueden usar, y es por razones geográficas, dependerá de la ubicación donde se necesite utilizar estos canales. En América no se pueden utilizar los canales 12, 13 y 14, en China además de estos no se puede usar el canal 1. En cambio, en Japón se pueden usar todos los canales, al contrario que Israel que es el dominio regulador que menos canales permite solo se pueden usar los canales del 3 al 9.

A diferencia de la banda 2.4GHz, en la banda 5GHz cada canal tiene un ancho de 20MHz con una separación de 5 MHz, la distancia entre canales sin superposición pasa de 4 en 2.4GHz a 3 en 5GHz, demás esta banda posee 23 canales sin solapamiento que son legales utilizarlos, existen otros canales que no generan solapamiento pero que son ilegales ya que están destinado para el uso militar.

Un punto muy importante en los dispositivos que usan esta banda es que solo permiten configurarlos en los canales que no se superponen, además tiene como curiosidad esta banda que el primer canal configurable es el 36 y el ultimo el 165.

2.2.6 Modos WiFi

Todos los dispositivos WiFi pueden ser configurados para que actúen de una forma u otra dependiendo de qué es lo que se necesite, cabe aclarar que no todos los protocolos que existen pueden operar en todos los modos que se describirán a continuación. Estas configuraciones son: modo master o de infraestructura, modo repetidor y modo ad-hoc.

2.2.6.1 Modos Master o de infraestructura

Una tarjeta en modo master ofrece un servicio de red que parece ser un punto de acceso (AP) tradicional, dicho de otra forma, crea una red inalámbrica que por lo general permite la conexión a internet a los usuarios de esta. Para que este modo pueda operar se le debe configurar un canal y un nombre de red, dicho nombre es conocido como SSID. La asignación del canal debe de ser asignado procurando que no se vea afectado por otros dispositivos que estén actuando con este mismo modo y estén utilizando el mismo canal de comunicación o uno que lo traslape. Otra característica de este modo es que no pueden comunicarse con los demás dispositivos configurados con este mismo modo, en cambio brindan conexiones a los dispositivos configurados en modo administrativo.

2.2.6.2 Modo repetidor

Se utiliza para ampliar el alcance de una red inalámbrica.

2.2.6.3 Modo ad-hoc

Una red ad-hoc inalámbrica es aquella en las que no hacen falta nodos especiales que encaminen y gestionen el tráfico, sino que cada nodo tiene la capacidad de reenviar paquetes dirigidos a otros nodos de la red. Dicho de otra forma, se puede tener una red de dispositivos que se comuniquen sin necesidad de enrutadores o puntos de acceso. Este es el modo fundamental para la creación de las redes Mesh que tanto se mencionan en este proyecto, pero las cuales serán explicadas más adelante.

2.2.7 Seguridad

De manera informal, se define la seguridad en función de los siguientes términos:

- Integridad: Se dice que se mantiene la integridad de una comunicación cuando no es posible realizar modificaciones o cambios en los datos transmitidos o recibidos por personal no autorizado. En ocasiones, el concepto también se denomina inviolabilidad.

- Confidencialidad: El contenido de las comunicaciones sólo puede ser conocido por el emisor y el receptor.
- Disponibilidad: El soporte de los sistemas de comunicaciones han de estar siempre disponibles, en la medida de las posibilidades técnicas, para su uso.

Las redes WLAN han adolecido de fallos en estos tres puntos y fundamentalmente en el tema de la privacidad, dado que el contorno de la red no está perfectamente definido y cualquier intruso puede realizar una escucha pasiva de los datos en circulación y analizarlos.

En cuanto a la confidencialidad, el punto débil se encuentra en los ordenadores conectados a la red inalámbrica que pueden ser atacados, si no disponen de las adecuadas medidas de seguridad, y comprometer la información recibida.

Dos son los problemas principales a resolver:

- Autentificación y control de acceso.
- Cifrado.

Autentificación y control de acceso:

- SSID (Service Set Identifier): Basado en contraseña (inútil).
- Seguridad por restricción de direccionamiento MAC: Sólo los terminales cuya MAC se encuentre en una determinada lista pueden acceder a la red (inútil).
- Contraseñas no estáticas (periódicas o de un solo uso): Absolutamente inútil.

Cifrado:

- WEP: Emplea el algoritmo de cifrado RC4, muy robusto y eficiente. El problema de WEP proviene de la debilidad de las claves que son de 64, 128 y 156 bits que pueden descubrirse con relativa facilidad por cualquier aficionado.
- WPA: Basado en una adaptación que hizo Microsoft del cifrado TKIP y que denominó SSN, fue adoptado dentro de 802.1x.
- WPA2: Evolución del anterior y que proporciona un cifrado AES. Desgraciadamente consume muchos recursos de computación.

2.2.8 Componentes de la red

La ruta que toma un mensaje desde el origen hasta el destino puede ser tan sencilla como un solo cable que conecta una computadora con otra o tan compleja como una red que literalmente abarca el mundo. Esta infraestructura de red es la plataforma que da soporte a la red. Proporciona el canal estable y confiable por el cual se producen las comunicaciones.

La infraestructura de red contiene tres categorías de componentes de red:

- Dispositivos
- Medios
- Servicios

Los dispositivos y los medios son los elementos físicos o el hardware, de la red. Por lo general, el hardware está compuesto por los componentes visibles de la plataforma de red, como una computadora portátil, una PC, un switch, un router, un punto de acceso inalámbrico o el cableado que se utiliza para conectar esos dispositivos. A veces, puede que algunos componentes no sean visibles. En el caso de los medios inalámbricos, los mensajes se transmiten a través del aire mediante radio frecuencias invisibles u ondas infrarojas.

Los componentes de red se utilizan para proporcionar servicios y procesos, que son los programas de comunicación, denominados “software”, que se ejecutan en los dispositivos conectados en red. Un servicio de red proporciona información en respuesta a una solicitud. Los servicios incluyen muchas de las aplicaciones de red comunes que utilizan las personas a diario, como los servicios de hosting de correo electrónico y web hosting.

Los procesos proporcionan la funcionalidad que direcciona y traslada mensajes a través de la red. Los procesos son menos obvios para nosotros, pero son críticos para el funcionamiento de las redes.

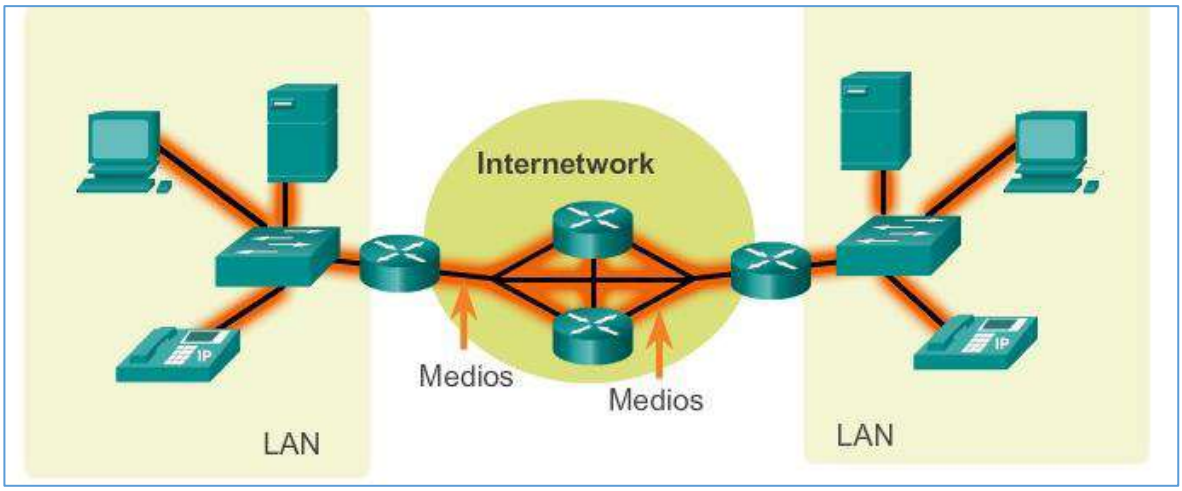


Figura 6: Arquitectura de una red LAN

2.2.8.1 Dispositivos finales

Los dispositivos de red con los que las personas están más familiarizadas se denominan “dispositivos finales” o “hosts”. Estos dispositivos forman la interfaz entre los usuarios y la red de comunicación subyacente.

Algunos ejemplos de dispositivos finales son:

- Computadoras (estaciones de trabajo, computadoras portátiles, servidores de archivos, servidores web).
- Impresoras de red.
- Teléfonos VoIP.
- Terminales de TelePresence.
- Cámaras de seguridad.
- Dispositivos portátiles móviles (como smartphones, tablet PC, PDA y lectores inalámbricos de tarjetas de débito y crédito, y escáneres de códigos de barras).

Un dispositivo host es el origen o el destino de un mensaje transmitido a través de la red, tal como se muestra en la Figura 7. Para distinguir un host de otro, cada host en la red se identifica por una dirección. Cuando un host inicia la comunicación, utiliza la dirección del host de destino para especificar a dónde se debe enviar el mensaje.

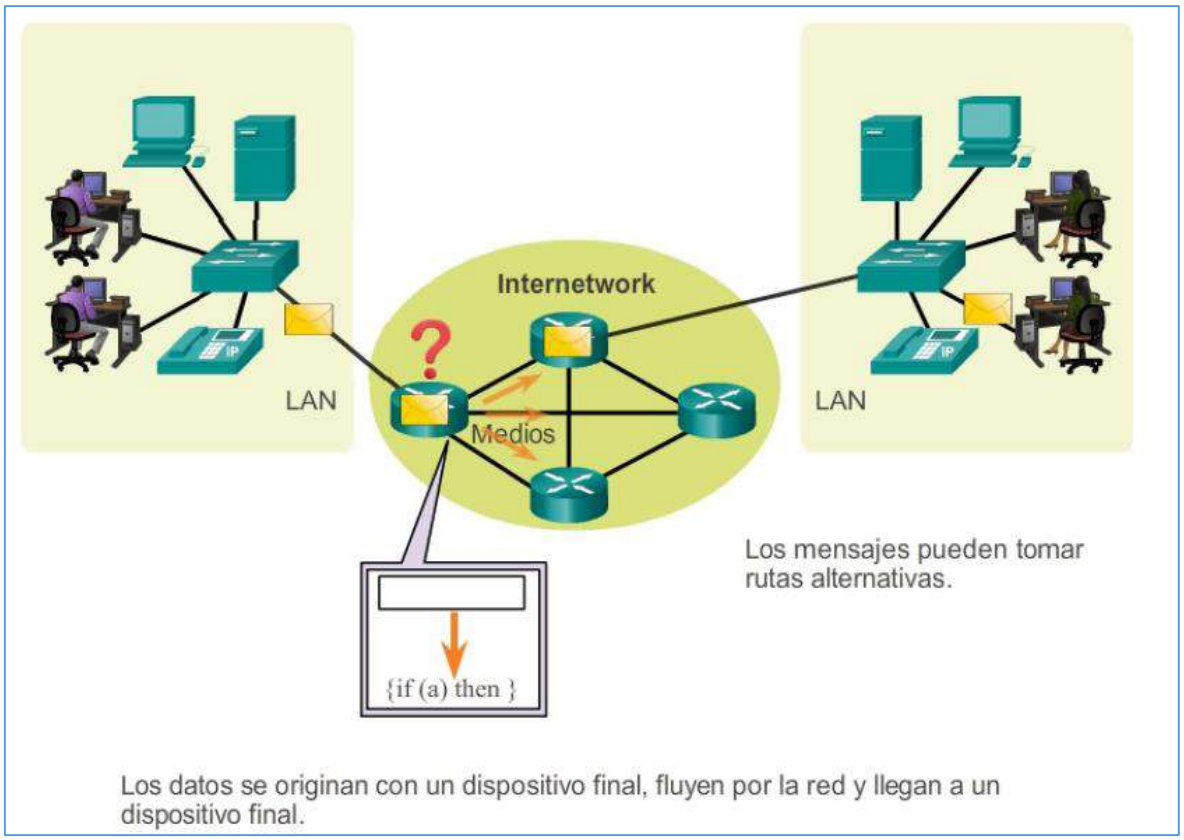


Figura 7: Arquitectura de una red LAN con dispositivos finales

2.2.8.2 Dispositivos de red intermediarios

Los dispositivos intermediarios interconectan dispositivos finales. Estos dispositivos proporcionan conectividad y operan detrás de escena para asegurar que los datos fluyan a través de la red, como se muestra en la Figura 8. Los dispositivos intermediarios conectan los hosts individuales a la red y pueden conectar varias redes individuales para formar una internetwork.

Los siguientes son ejemplos de dispositivos de red intermediarios:

- Acceso a la red (switches y puntos de acceso inalámbrico).
- Internetworking (routers).
- Seguridad (firewalls).

La administración de datos, así como fluye en la red, es también una función de los dispositivos intermediarios. Estos dispositivos utilizan la dirección host de destino, conjuntamente con información sobre las interconexiones de la red para determinar la ruta que deben tomar los mensajes a través de la red.

Los procesos que se ejecutan en los dispositivos de red intermediarios realizan las siguientes funciones:

- Volver a generar y transmitir las señales de datos.
- Conservar información acerca de las rutas que existen a través de la red y de internetwork.
- Notificar a otros dispositivos los errores y las fallas de comunicación.
- Dirigir los datos a lo largo de rutas alternativas cuando hay una falla en el enlace.
- Clasificar y dirigir los mensajes según las prioridades de calidad de servicio (QoS, Quality of Service).
- Permitir o denegar el flujo de datos de acuerdo con la configuración de seguridad.

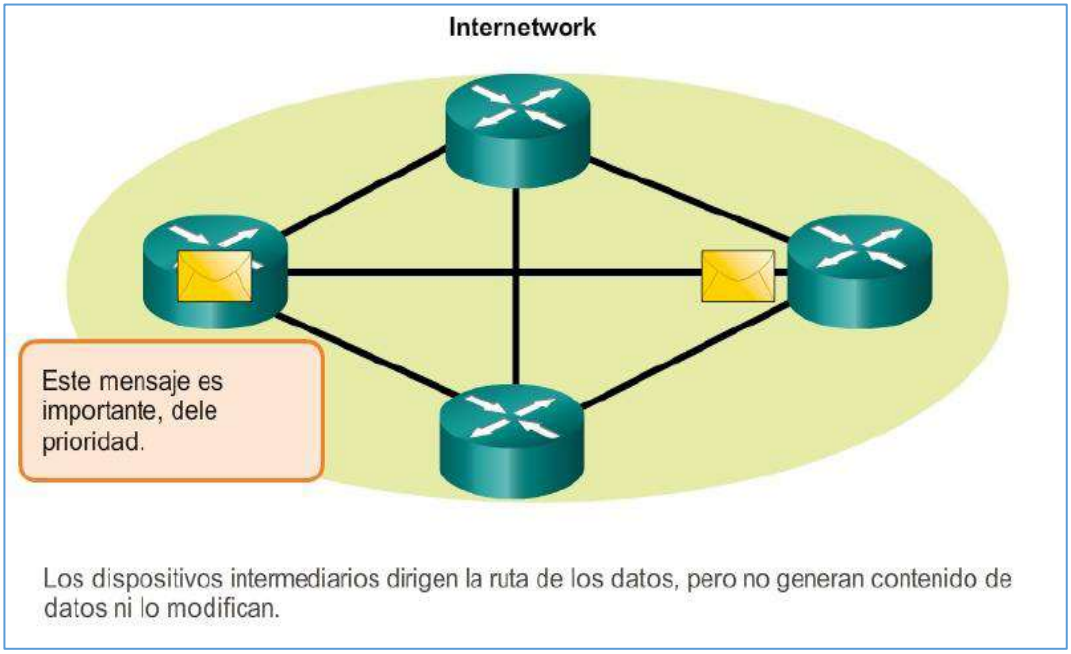


Figura 8: Dispositivos intermediarios

2.2.8.3 Medios de red

La comunicación a través de una red es transportada por un medio. El medio proporciona el canal por el cual viaja el mensaje desde el origen hasta el destino.

Las redes modernas utilizan principalmente tres tipos de medios para interconectar los dispositivos y proporcionar la ruta por la cual pueden transmitirse los datos. Como se muestra en la figura 9, estos medios son los siguientes:

- Hilos metálicos dentro de cables.
- Fibras de vidrio o plástico (cable de fibra óptica).

- Transmisión inalámbrica.

La codificación de la señal que se debe realizar para que se transmita el mensaje es diferente para cada tipo de medio. En los hilos metálicos, los datos se codifican dentro de impulsos eléctricos que coinciden con patrones específicos. Las transmisiones por fibra óptica dependen de pulsos de luz, dentro de intervalos de luz visible o infrarroja. En las transmisiones inalámbricas, los patrones de ondas electromagnéticas muestran los distintos valores de bits.

Los diferentes tipos de medios de red tienen diferentes características y beneficios. No todos los medios de red tienen las mismas características ni son adecuados para el mismo fin. Los criterios para elegir medios de red son los siguientes:

- La distancia por la que los medios pueden transportar una señal correctamente.
- El entorno en el que se instalarán los medios.
- La cantidad de datos y la velocidad a la que se deben transmitir.
- El costo del medio y de la instalación.

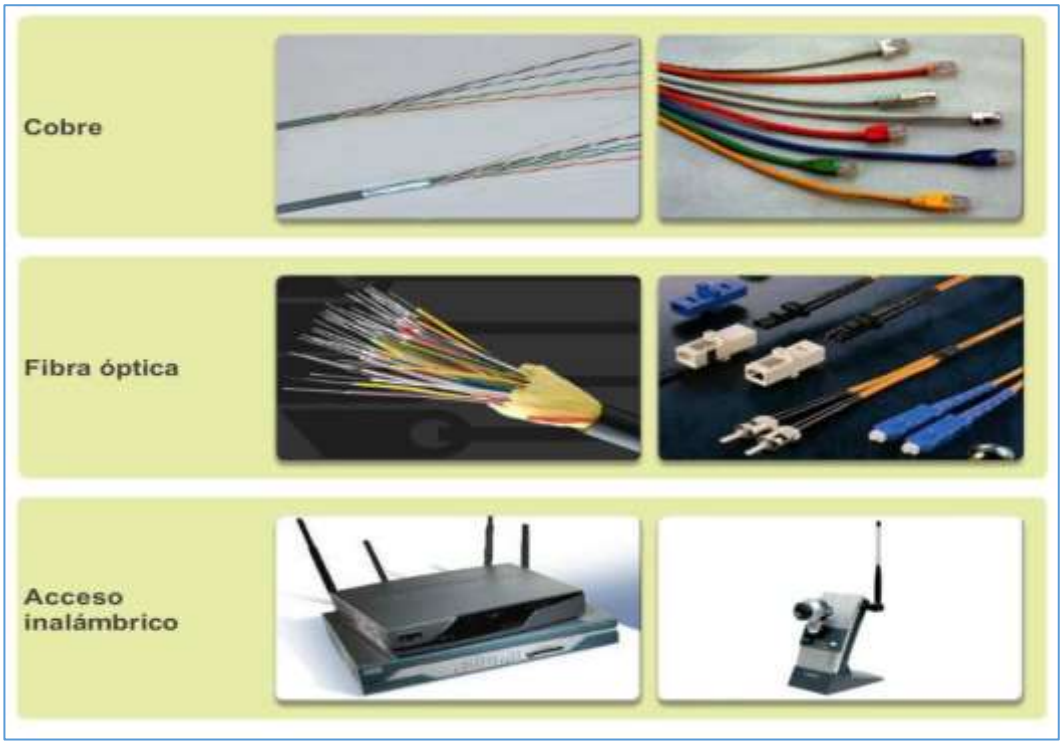


Figura 9: Diferentes medios de red

2.2.8.4 Representaciones de red

Para transmitir información compleja, como la presentación de todos los dispositivos y el medio en una internetwork grande, es conveniente utilizar representaciones visuales. Los diagramas permiten comprender fácilmente la forma en la que se conectan los dispositivos en una red grande. Estos diagramas utilizan símbolos para representar los diferentes dispositivos y conexiones que componen una red. Este tipo de representación de una red se denomina “diagrama de topología”.

Como cualquier otro lenguaje, el lenguaje de las redes se compone de un conjunto común de símbolos que se utilizan para representar los distintos dispositivos finales, dispositivos de red y medios, como se muestra en la Figura 10. La capacidad de reconocer las representaciones lógicas de los componentes físicos de red es fundamental para poder visualizar la organización y el funcionamiento de una red.

Además de estas representaciones, se utiliza terminología especializada al hablar sobre cómo se conectan estos dispositivos y los medios unos a otros. Algunos términos importantes para recordar son:

- Tarjeta de interfaz de red: una NIC, o adaptador LAN, proporciona la conexión física a la red para la PC u otro dispositivo host. Los medios que realizan la conexión de la PC al dispositivo de red se conectan en la NIC.
- Puerto físico: se trata de un conector o una boca en un dispositivo de red donde se conectan los medios a un host u otro dispositivo de red.
- Interfaz: puertos especializados en un dispositivo de internetworking que se conectan a redes individuales. Puesto que los routers se utilizan para interconectar redes, los puertos de un router se conocen como interfaces de red.

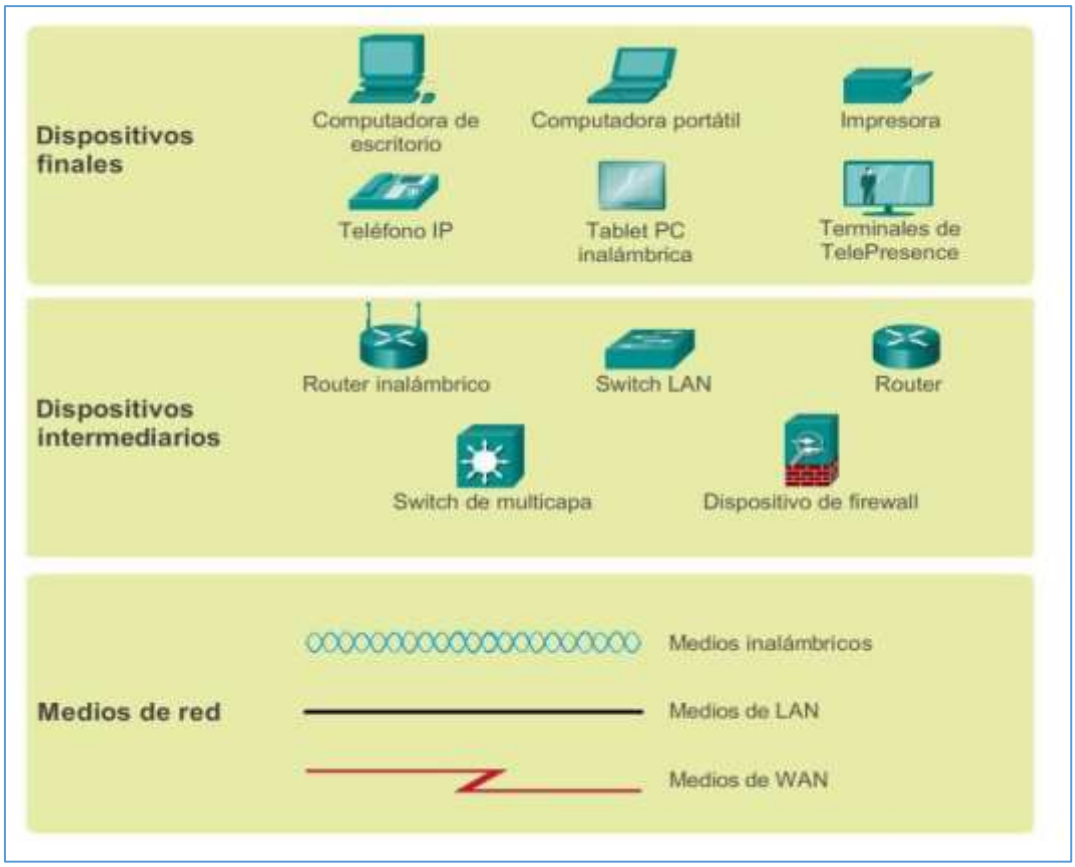


Figura 10: Representaciones de red

2.2.8.5 Conectores UTP

Los cables UTP se terminan generalmente con un conector RJ-45 especificado por el estándar ISO 8877. Este conector se utiliza para una variedad de especificaciones de capa física, una de las cuales es Ethernet.

Según las diferentes situaciones, es posible que los cables UTP necesiten armarse según las diferentes convenciones para los cableados. Esto significa que los hilos individuales del cable deben conectarse en diferente orden para distintos grupos de pines en los conectores RJ-45.

A continuación se mencionan los principales tipos de cables que se obtienen al utilizar convenciones específicas de cableado:

- Cable directo de Ethernet: el tipo más común de cable de red. Por lo general, se utiliza para interconectar un host con un switch y un switch con un router.

- Cable cruzado Ethernet: cable poco común utilizado para interconectar dispositivos similares. Por ejemplo, para conectar un switch a un switch, un host a un host o un router a un router.

En la figura 11, se muestra un ejemplo de un conector RJ-45 de un cable Ethernet.

Figura 11: Conectores RJ-45 para UTP

3 OBJETIVOS

Los objetivos del proyecto son los siguientes:

Poner en marcha una red MANET que será la encargada de transmitir niveles de potencia captados de los iBeacon y, a su vez, transmitir adecuadamente y de forma cifrada esta información a un sistema central de procesado de la información.

Los nodos de dicha red estarán basados en equipos Raspberry Pi 3 con sistema operativo Raspbian que es una variante del Linux Debian. La conexión de los nodos será utilizando la tecnología de la red WiFi.

Por otro lado, se va a configurar el protocolo OLSRD que será encargado de trabajar en forma distribuida para establecer las conexiones entre los nodos de la red ad-hoc (MANET).

Por último, se va a configurar en un nodo Raspberry el software base que nos permitirá desarrollar la aplicación basada en Bluetooth (en un caso sencillo, recibir la ráfaga de los datos procedentes de un iBeacons) y que además permita enviar los datos recibidos de los dispositivos Bluetooth a un sistema de proceso central (que puede estar en otro nodo Raspberry o bien puede ser un ordenador convencional). Para ello utilizaremos la tecnología denominada “Websocket” que será controlada, además de las comunicaciones Bluetooth mediante el software de desarrollo Nodejs; que no es más que un framework que nos permitirá desarrollar la aplicación en Javascript fuera de un navegador.

Mediante la realización de las mediciones de la red MANET, el sistema almacena la información detallada de los dispositivos, los niveles de potencia, así como otros datos relevantes, con ellos la aplicación realiza un preprocesamiento con un diagnóstico de situación de la red y proporciona recomendaciones para mejorarla.

4 MATERIALES Y MÉTODOS

4.1 DESCRIPCIÓN SOBRE DISPOSITIVOS UTILIZADOS EN LA RED MANET

En esta sección del documento se describe los elementos necesarios en base a los servicios que cada dispositivo pretende brindar, tanto a la hora de establecer una red de comunicación tipo Mesh como en la comunicación por Bluetooth.Estos dispositivos son: Raspberry Pi 3B e iBeacons.

4.1.1 Raspberry Pi 3B

Es una computadora de placa reducida que soporta los servicios que requiere el proyecto, ya que según las especificaciones que tiene puede adicionarse tanto como el hardware como el Software necesario. Tiene buenas prestaciones de procesamiento y ofrece flexibilidad en cuanto al Hardware y el Software que puede soportar. Un aspecto interesante es que dentro del Hardware adicional que soporta puede ser por conexión USB, por lo que se trata de Hardware bastante comercial del que se puede obtener con relativa facilidad los controladores y software necesario para operarlo. El hecho de que soporte diversos sistemas operativos Linux ofrece mucha versatilidad, porque se puede elegir dentro de las distribuciones soportadas la que mejores prestaciones ofrezca para el soporte de los servicios.

Figura 12: Placa base Raspberry Pi 3B

4.1.1.1 Especificaciones técnicas del dispositivo

Raspberry Pi 3B que se usara para el desarrollo de este proyecto se listan las especificaciones de hardware con las cuales cuentan esta placa:

- Procesador a 1.2GHz de 64 bits con cuatro núcleos ARMv8.
- Memoria RAM de 1GB.
- Bluetooth 4.1.

- Bluetooth Low Energy (BLE) ó 4.0
- 4 puertos USB.
- 40 pines GPIO.
- Puerto Full HDMI.
- Puerto Ethernet.
- Conector compuesto de audio y vídeo de 3.5mm.
- Interfaz de cámara (CSI).
- Interfaz de pantalla (DSI).
- Ranura para tarjetas microSD (ahora push-pull en lugar de push-push).
- Núcleo de gráficos VideoCore IV 3D.
- Dimensiones de placa de 8.5 por 5.3 cm.
- La Raspberry Pi 3 tiene un factor de forma idéntica a la anterior Pi 2 (1 y Pi Modelo B +) y tiene una compatibilidad completa con Frambuesa Pi 1 y 2.

4.1.2 iBeacons

La traducción de la palabra “beacon” sería baliza o faro, la letra i es característica de los productos de Apple para referirse a la tecnología que hace uso de estas balizas.

Se trata de una tecnología desarrollada por Apple a mediados de 2013. Aprovechando el lanzamiento de sistema operativo IOS7 para sus dispositivos móviles y utilizada para posicionamiento en interiores IPS (Indoor Positioning System).

La tecnología radio de base empleada es Bluetooth 4.0 o LE (Low Energy) disponible en todos los Raspberry Pi 3 modelo B.

Son dispositivos receptores donde no se aplica ningún tipo de patente; de hecho, ya existe para terminales con sistema operativo Android.

Los beacons son dispositivos autónomos, alimentados por batería tipo moneda y de muy pequeño tamaño y se supone que debajo precio, aunque actualmente no tanto debido a que son muy pocos los fabricantes actuales y aunque dependen de las unidades solicitadas, su precio suele rondar los 30 euros por unidad.

Figura 13: Imagen de un faro iBKS 105 beacon Bluetooth Low Energy.

4.1.2.1 Funcionamiento

Un beacon es una radiobaliza Bluetooth 4.0 que emite ráfagas de datos en tiempo programable. Evidentemente no es un dispositivo interactivo, en el sentido de que no dispone de un canal de radio de regreso y, por tanto, no permite que un dispositivo receptor interacciones con él en absoluto. El beacon se limita a emitir los tres siguientes parámetros:

UUUID:

- Cadena de 32 dígitos hexadecimales, o sea, 128 bits.
- Este parámetro sería, en principio, el identificador de la empresa que utiliza el beacon. Es de suponer que todos los beacons pertenecientes a la misma empresa emitan este mismo parámetro. Ejemplo típico sería: 41fffb422be45480be0e29de19634b79.

Mayor:

- Cadena de 4 dígitos hexadecimales.16 bits.
- Este parámetro es un digito de 16 bits, lo que nos da 65.536 posibilidades distintas por cada UUUID.
- Normalmente, este digito indicaría el grupo al que pertenece este beacon en particular.

Minor:

- Cadena de 4 dígitos hexadecimales. 16 bits
- Este parámetro es un digito de 16 bits, lo que nos da 65.536 posibilidades distintas por cada Mayor.
- Normalmente, este digito indicaría el identificador de un beacon particular dentro de un grupo de beacons.

4.1.2.2 Especificaciones eléctricas y mecánicas del dispositivo.

Esta ilustración contiene especificaciones eléctricas y mecánicas y de software para iBKS 105.

Dimensions	Ø52.6 x 11.3 mm	**Case material**	ABS
Weight	24g	**Case finish**	Matte white
Core	Nordic nRF51822	**Fixing method**	Double side sticker
Radio Protocol	Bluetooth® Low Energy	**Operating Temperature**	-25 to +60°C
Distance Range	Up to 50m	**Storage Temperature**	0 to +35°C
Battery	Coin Cell CR2477 3V – 1000mAh	**Beacon Protocols**	iBeacon Eddystone: UID, URL, TLM & EID
Optional Sensors	Hall Accelerometer	**Firmware Update**	OTA (Over The Air)
Idle Current Consumption	2.4µA	**Certifications**	CE, FCC, IC, Anatel

Figura 14: iBKS 105 Datasheet.

La estimación de la vida útil de la batería se aplica desde la versión de firmeware EDSTID V5.2016.06.29.1 en adelante.

Se observa que todos los valores que se muestra en la hoja de características son solo una estimación. La duración real de la batería depende de la frecuencia de ráfaga.

4.2 INSTALACIÓN Y PUESTA EN SERVICIO DE UNA RED MESH AD-HOC CON NODOS BASADOS EN MINIORDENADORES RASPBERRY PI 3B

En este capítulo se describe el proceso de instalación, configuración y puesta en servicio de una red mesh ad-hoc mediante nodos Raspberry pi; concretamente, el modelo 3 B.

4.2.1 Redes ad-hoc

Recordemos que una red ad-hoc está formada por una serie de nodos que se comunican entre sí de manera inalámbrica sin necesidad de un punto de acceso central; no existe, por tanto, un nodo privilegiado (como sería un Punto de Acceso WiFi), sino que todos los nodos tienen la misma jerarquía.

Una red de este tipo está constituida por una única célula (que puede ser extendida tanto como deseemos) que configura el medio de transmisión que comparten todos los nodos; por supuesto, la célula tiene un solo canal y un solo nombre.

El ancho de banda del canal es compartido por todos los nodos y existen, por tanto, mecanismos de contención en el acceso al medio que impiden que un nodo lo utilice si otro lo está ocupando. Al no existir retransmisión en un Punto de Acceso, las comunicaciones serán a la máxima velocidad que permita el canal.

Al compartir todos los nodos el mismo canal, obviamente, todos habrán de pertenecer a la misma subred; habremos de ser cuidadosos, por tanto, en asignar a cada nodo una dirección diferente en la red de modo fijo (nos referimos, naturalmente, a las interfaces de los nodos directamente conectadas a la red ad-hoc.

Hasta aquí, esta es la definición de una red simple ad-hoc, que podemos formar entre equipos de ordenadores que harían de nodos en la red y que nos permitirían compartir archivos, etc. Sin más problema.

El inconveniente principal que presenta una red de este tipo es que un nodo sólo puede alcanzar a aquellos otros que se encuentren dentro de su propia zona de cobertura. Es lógico, dado que el receptor del nodo no podrá captar emisiones

procedentes de más allá de su zona (no captaremos nada por debajo de la sensibilidad del receptor) y esto es así por mucho que extendamos la cobertura general añadiendo células.

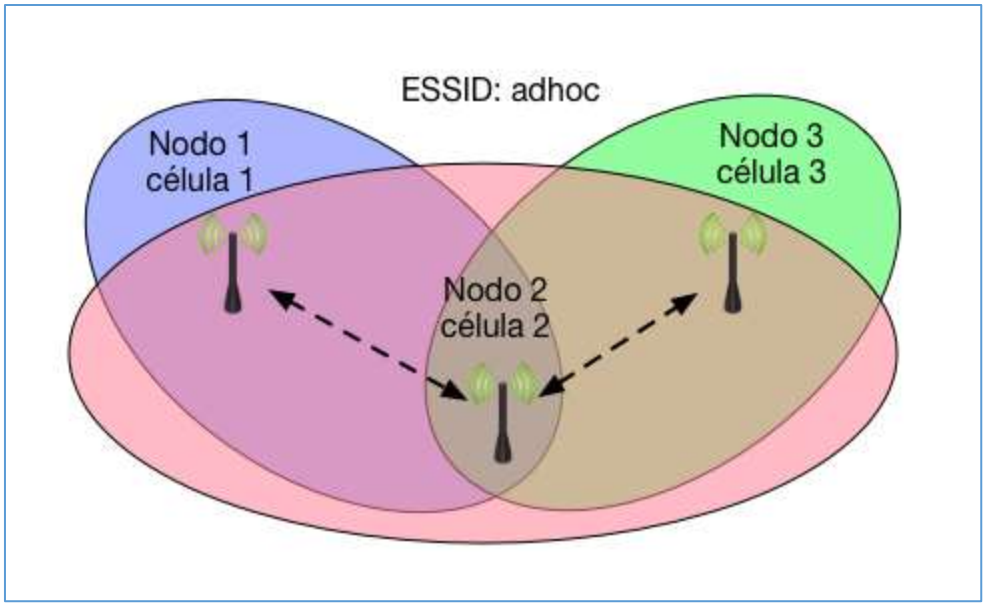


Figura 15: Red ad-hoc disjunta.

En la Figura anterior tenemos la representación de una red Ad-hoc disjunta, en la que aunque las tres zonas de cobertura (células) generadas por los nodos 1 (azul), 2 (rojo) y 3 (verde) se superponen, sólo podemos tener comunicación entre el nodo 3 con el 2 y el nodo 1 con el 2, siendo imposible la comunicación entre los nodos 1 y 3 dado que el nodo 1, aunque superpone su área de cobertura con la del nodo 3, no se encuentra también dentro de dicha área.

Recordemos que el área de cobertura sería el lugar geográfico de todos los puntos en los que la potencia recibida es mayor que la sensibilidad del receptor, por lo que es imposible hacer que el área de cobertura completo de la red (que todos los nodos puedan comunicarse entre sí) sea mayor que la más pequeña de todas las áreas de cobertura generadas por un único nodo. El nodo con peores características será el que nos limite el tamaño máximo de la red que podemos crear.

Por supuesto, si configurásemos el nodo 2 como un bridge (lo que no resulta tan sencillo de realizar), tendríamos continuidad en las capas 1 y 2 de la red entre los nodos 1 y 3 y sería posible la comunicación entre ambos, pero esto ya no es una red Ad-hoc pura, en el sentido de que uno de los nodos es diferente de los demás, además de que la situación relativa entre los nodos debería de mantenerse fija o de lo contrario

perderíamos la continuidad. No es aplicable, por tanto, a redes formadas por dispositivos móviles.

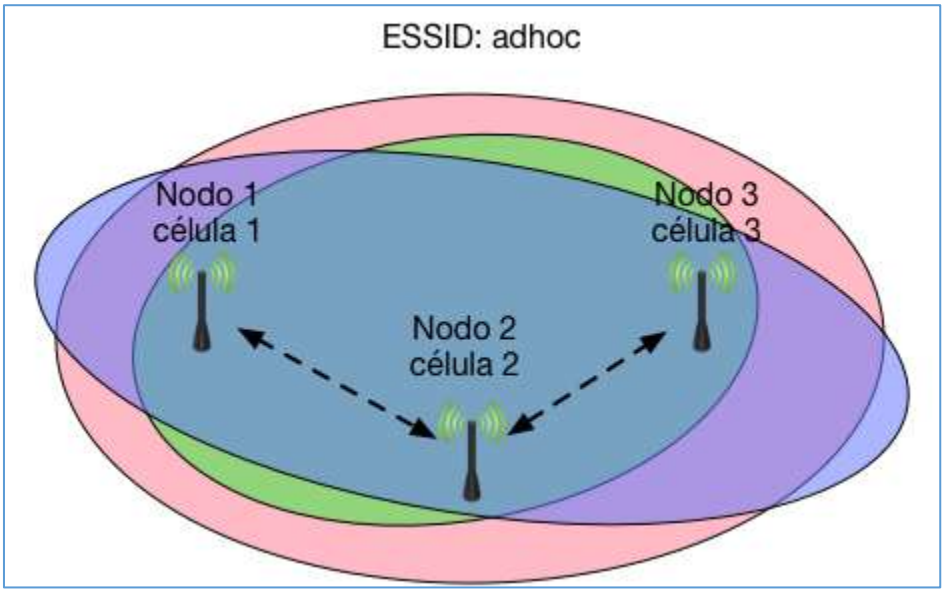


Figura 16: Red ad-hoc disjunta con áreas de cobertura modificadas.

En la Figura anterior hemos modificado las áreas de cobertura y hemos definido que el área del nodo 2 (rojo) sea la más extensa, seguida del área creada por el nodo 1 (azul) y la más pequeña es la del nodo 3 (verde). Como podemos apreciar, si quisiéramos añadir un nuevo nodo a la red, no tenemos otra opción que colocarlo dentro del área verde, es decir, el área más pequeña.

Conclusión: El peor dispositivo define el tamaño máximo que puede tener la red (en lo que se refiere a tamaño geográfico, claro).

4.2.2 Conceptos generales de las Redes Mesh

Una red ad-hoc (Mobile Ad-hoc Network o Manet) (Véase la Figura 16) es un conjunto de nodos o hosts que se comunican entre sí mediante enlaces wireless, sin la necesidad de una infraestructura de red fija. Cada nodo actúa como router y va encaminando los distintos paquetes entre los diferentes terminales, sin la necesidad de que exista un alcance directo entre la fuente y el destino.

Este tipo de red es autónoma entre terminales que pueden moverse libremente.

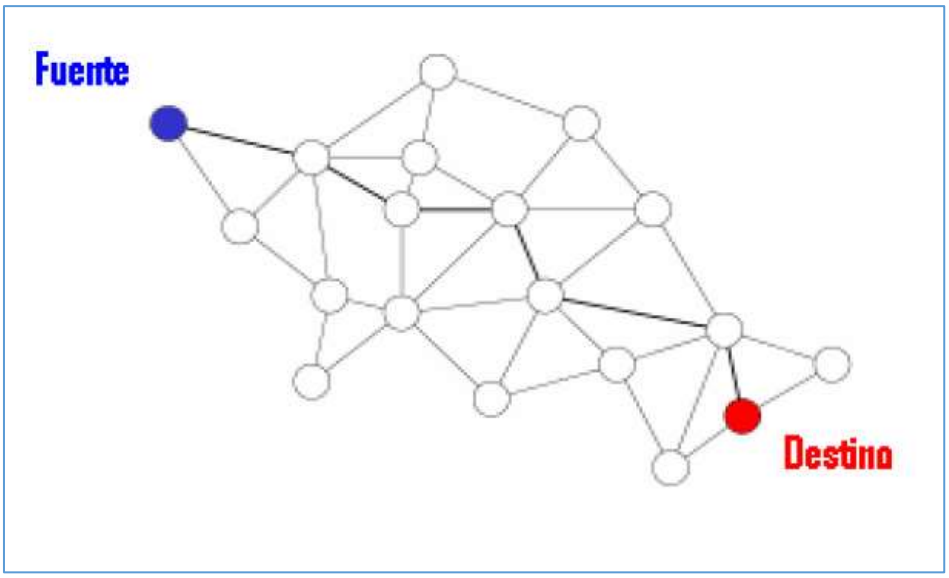


Figura 17: Red ad hoc

Los nodos de una Manet utilizan tablas de encaminamiento para organizar los saltos entre los nodos intermedios y así poder enviar los paquetes. Estas tablas de encaminamiento se deben actualizar con mucha frecuencia ya que la red es móvil y van cambiando de posición constantemente.

4.2.3 Características de las Redes Mesh

Las principales características de las redes ad hoc son:

- Los terminales autónomos. En este tipo de redes cada nodo es autónomo pudiendo funcionar como encaminador (router).
- Funcionamiento distribuido. Una Manet puede formarse sin la necesidad de infraestructura y al mismo tiempo conectarse a una para mejorar su capacidad. Por eso todos los terminales se encargan del control y la gestión de la red.
- Encaminamiento multisalto. Los paquetes realizan uno o varios saltos para llegar de la fuente al destino.
- Topología de red dinámica. Todos los nodos de la red son terminales móviles que pueden moverse libremente cambiando constantemente su arquitectura de red. Esto introduce una cierta complejidad al realizar el encaminamiento.
- Enlace fluctuante. Al tratarse de conexiones inalámbricas utilizan como medio de transmisión el aire. Esto supone una alta tasa de errores al tener las limitaciones del ruido, atenuación, ancho de banda limitado.

- Terminales ligeros. La capacidad de los terminales acostumbra a ser baja al igual que el de las baterías.

4.2.4 Clasificación de las Redes Mesh

Las redes Mesh pueden ser cableadas, (lo que ahorra mucho trabajo a los administradores de red), inalámbricas o mixtas. En este proyecto, nos vamos a centrar, en un primer paso en las inalámbricas, y en un segundo paso en las mixtas.

4.2.4.1.1 Redes Mesh inalámbricas

El medio de transmisión entre los nodos es el espacio aéreo. Este tipo de redes (su concepto) es independiente de la tecnología inalámbrica empleada, así podrían ser: Bluetooth, WiFi, Zigbee, Direct WiFi, etc. Siempre que dicha tecnología cuente en la capa de red con los pertinentes mecanismos de acceso al medio compartido.

Los diversos nodos que componen la red se comunican entre sí en función de la tecnología, y basta con ponerlos en marcha (una vez adecuadamente configurados) para que la red encuentre el camino más adecuado entre todos los nodos.

4.2.4.1.2 Redes Mesh mixtas

Normalmente, una red Mesh estará formada por nodos que pueden, a su vez, tener accesos a redes cableadas (Internet, redes internas, etc.). Las rutas de las redes cableadas también habrán de incorporarse a las tablas generales para que cualquier nodo inalámbrico pueda acceder (o no) a las redes cableadas.

4.2.4.1.3 Redes Mesh ad-hoc basadas en la especificación 802.11

Son redes Mesh WiFi que, evidentemente, están configuradas en modo ad-hoc. Hemos visto anteriormente que, si disponemos de una única célula, la cobertura de la red está limitada al tamaño de célula del peor dispositivo; en este caso no tiene sentido hablar de encaminamiento, dado que sólo existe una célula.

Si deseamos crear una red funcional, hemos de superponer el número adecuado de células que nos permitan cubrir una determinada área de interés.

En cada una de las células habrán de existir, al menos, dos nodos de modo que sea posible acceder desde cualquier nodo de la red a cualquier otro, independientemente del área de cobertura o célula en la que se encuentre.

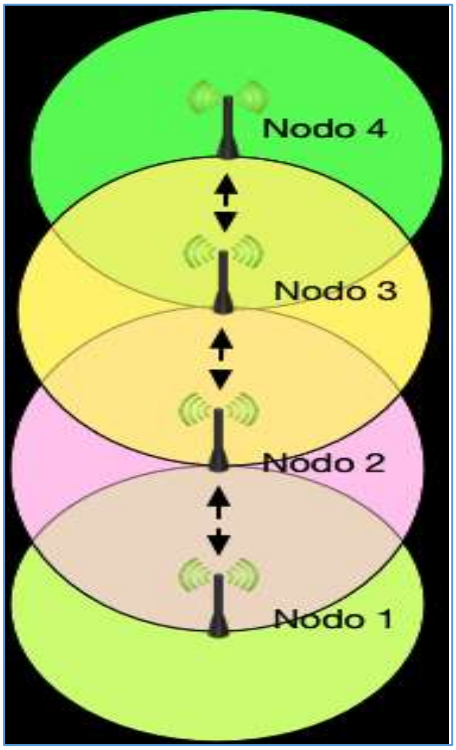


Figura 18: Red ad-hoc lineal

En la figura anterior podemos apreciar la arquitectura de una red lineal formada por cuatro nodos, en la que se cumplen la condición de que cada nodo puede acceder directamente a sus dos adyacentes, pero no más allá. Así, por ejemplo, el Nodo 1 puede acceder sólo al Nodo 2 y el Nodo 2 puede acceder al Nodo 1 y al Nodo 3 y así sucesivamente.

Es evidente que, en tal red, el Nodo 1 no puede acceder de modo directo al Nodo 3, sino sólo a través del Nodo 2 que debería entonces de comportarse como un Router.

De esta manera, cuando el Nodo 1 desee enviar una trama al Nodo 3, utilizará al Nodo 2 como Gateway para alcanzarlo.

En principio, bastaría con configurar las rutas en la red. Sin embargo, este procedimiento tiene sólo sentido cuando la red es absolutamente estática y además no demasiado complicada (en la anterior figura, sólo existe un camino para ir del Nodo 1 al Nodo 4).

En términos generales, las redes tendrán una topología mucho más compleja que incorporará, seguramente, subredes cableadas, etc.

También habremos de considerar la posibilidad de que los nodos que conforman la red sean móviles, con lo que no tiene sentido definir rutas estáticas en la red.

Dado que se trata de una red ad-hoc, el ESSID es idéntico para todas las células y es claro también que cada nodo habrá de tener una IP diferente. Por supuesto, los nodos adyacentes habrán de pertenecer a la misma subred (en otro caso no podrían encontrarse) pero dado que en una red móvil no sabemos cómo se van a redistribuir los nodos, la adyacencia es una propiedad que depende del tiempo. Como conclusión, todos los nodos deben de pertenecer a la misma subred.

4.2.5 Instalar sistema operativo (Raspbian NOOBS)

Recursos necesarios:

- Tarjeta micro SD: Es recomendable una tarjeta clase 10 de al menos 8 GB
- Una computadora con lector de tarjeta micro SD y sistema operativo Linux (Ubuntu, o Debian), para la instalación de la imagen de sistema operativo en computadoras con Windows ver Anexo1. Si no dispone de lector de tarjeta micro SD utilice un adaptador de tarjetas SD a USB.

Software necesario:

- Archivo con la imagen Raspbian NOOBS.

Ingresamos a la siguiente URL: https://www.raspberrypi.org/downloads/noobs/ y descargamos al archivo de la imagen de Raspbian en la computadora en un directorio al cual se puede acceder fácilmente.

Todo el proceso de instalación se explica con más detalle en el anexo1, por lo cual visualizamos la siguiente figura.

Figura 19: Interfaz gráfica de Raspbian NOOBS

4.2.6 Preparación para trabajar con SSH

Las configuraciones siguientes, se pueden realizar directamente en cada Raspberry (directamente desde el teclado conectado al dispositivo y un monitor asociado al mismo) o bien desde un acceso remoto, en caso de que se quiera realizar las configuraciones directamente saltemos este paso, para ejecutar un comando basta con presionar enter desde la terminal.

Para trabajar más cómodamente con los dispositivos se recomienda habilitar y configurar el SSH.

- Identificar la red LAN: Identificar la red, la máscara y el Gateway de red en la cual está desarrollando el prototipo.
- Configurar estáticamente la interfaz eth0 de la Raspberry: Modificando el archivo interfaces dentro del cual están las configuraciones de las interfaces.
- Reiniciar el servicio de Networking de la Raspberry para aplicar los cambios realizados.

Realizar los pasos anteriores con cada una de las Raspberries asignando una IP distinta a cada una, ya que cada una se identificará con esa IP y es por medio de la cual será establecida la conexión SSH; a continuación se muestran los pasos realizados para la configuración de este prototipo, teniendo en cuenta que las direcciones IP utilizadas para las interfaces eth0 son del ambiente de desarrollo de este grupo de trabajo de graduación.

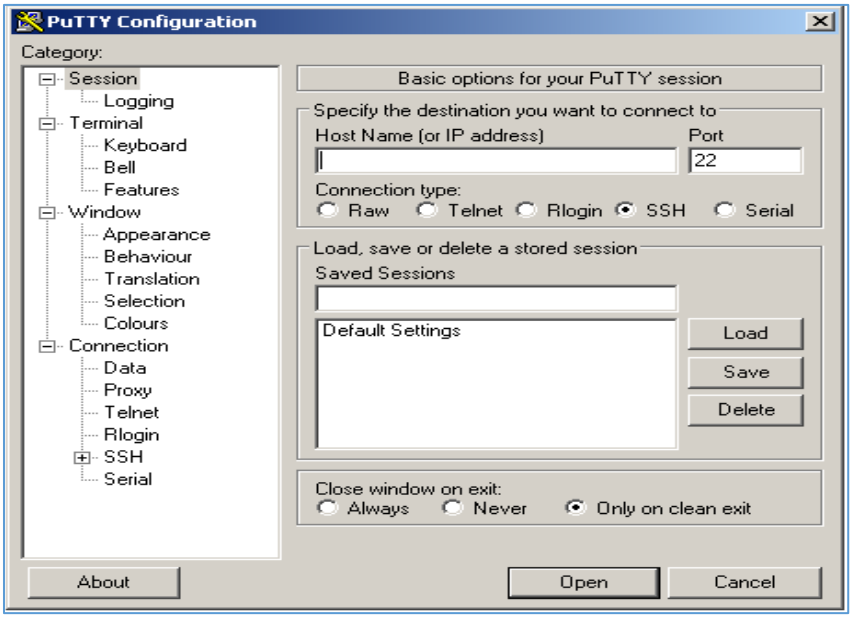


Figura 20: Programa de emulación de terminales (PuTTY)

Existen varios programas excelentes de emulación de terminales disponibles para conectarse a un dispositivo de red mediante una conexión serial por un puerto de consola o mediante una conexión Telnet o SSH

4.2.7 Configuración

Configuración de una red mesh ad-hoc y protocolo dinámico de encaminamiento OLSRD con tres nodos Raspberry pi. El nodo 1 está conectado mediante la red ethernet cableada a un Router con acceso a Internet y dirección 192.168.12.254.

Las direcciones ethernet de cada uno de los nodos pertenecen a redes distintas (192.168.12.0 para el nodo de acceso y 192.168.2.0 y 192.168.3.0 para los siguientes nodos), mientras que las direcciones WLAN de las mismas pertenecen a la misma red (192.168.1.0).

4.2.8 Direccionamiento de la red

El direccionamiento de la red es el que se indica a continuación

Nodo 1 (conectado a un Router con acceso a internet): Rasp1

- Eth0: 192.168.12.20
- Wlan0: 192.168.1.1

Nodo 2: Rasp2

- Eth0: 192.168.2.1
- Wlan0: 192.168.1.2

Nodo 3: Rasp3

- Eth0: 192.168.3.1
- Wlan0: 192.168.1.3

Tal como podemos observar nosotros hemos decidido a trabajar con direcciones IP de clase C para más comodidad.

- Queremos recordar las direcciones IP clase C, las cuales son utilizadas en pequeñas empresas, como instituciones educativas, el rango de la IP clase C empieza en 192 a 223, la máscara de subred es la 255.255.255.0.
- Se puede configurar 2^{21} (2097152) redes, cada una de ellas con 2^{8}-2(254) terminales o Host. En esta red los primeros octetos identifican la red mientras que el ultimo octeto identifica al terminal o Host quedando Red.Red.Red.Host

4.2.9 *Configuración básica en todos los nodos*

El procedimiento más sencillo consiste en instalar y configurar el nodo 1 para que se conecte a Internet, actualizar e instalar protocolo OLSRD. Después copiamos la tarjeta micro SD y para configurar los demás nodos.

El usuario y contraseña inicial son:

login: pi

password: raspberry

Por otro lado, se anexan los modelos de archivos de configuración, ya preparados, para el Nodo 1 que se conecta a Internet y el resto de nodos, estos archivos son:

- **/etc/hostname**
- **etc/hosts**
- **etc/init.d/inicio_olsrd**
- **etc/network/interfaces**
- **etc/olsrd/olsrd.conf**
- **/etc/sysctl.conf (no necesario en el resto de nodos)**

Una vez instalado el sistema base, mediante el mandato **sudo raspi-config** configuraremos el espacio de almacenamiento para ocupar toda la tarjeta micro SD.

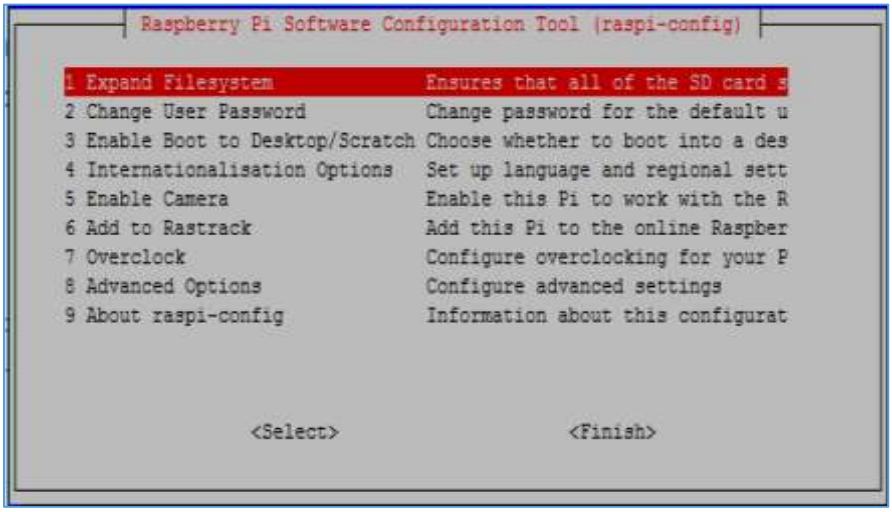


Figura 21: Interfaz de configuración de Raspberry

Entramos en la opción expand filesystem, al acceder el sistema permitirá utilizar todo el espacio de la memoria SD, al realizarlo se mostrará una pantalla como la que se muestra en la figura.

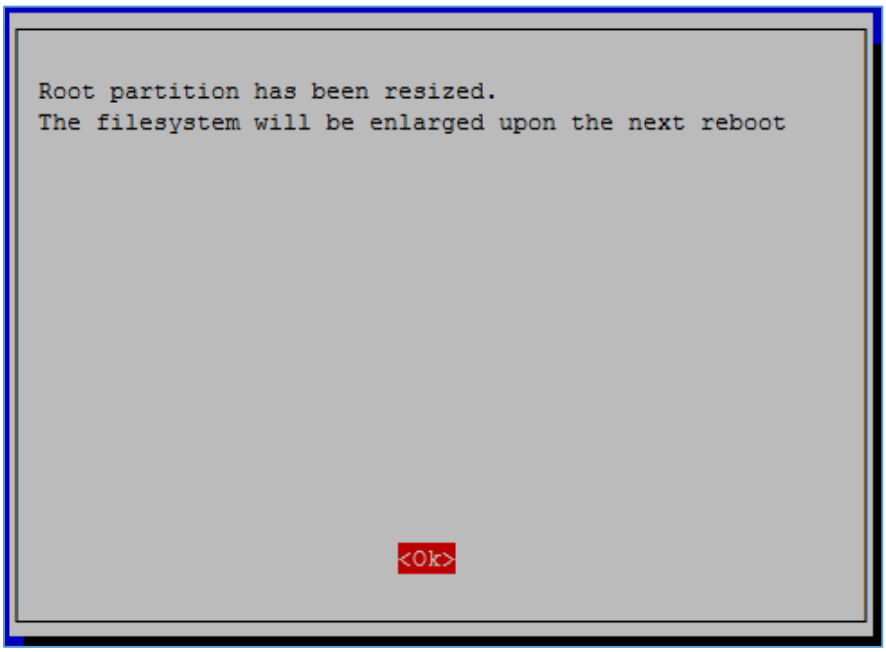


Figura 22: Configuración de expansión de disco.

Configuramos también el idioma del sistema y el teclado mediante la aplicación de la siguiente figura.

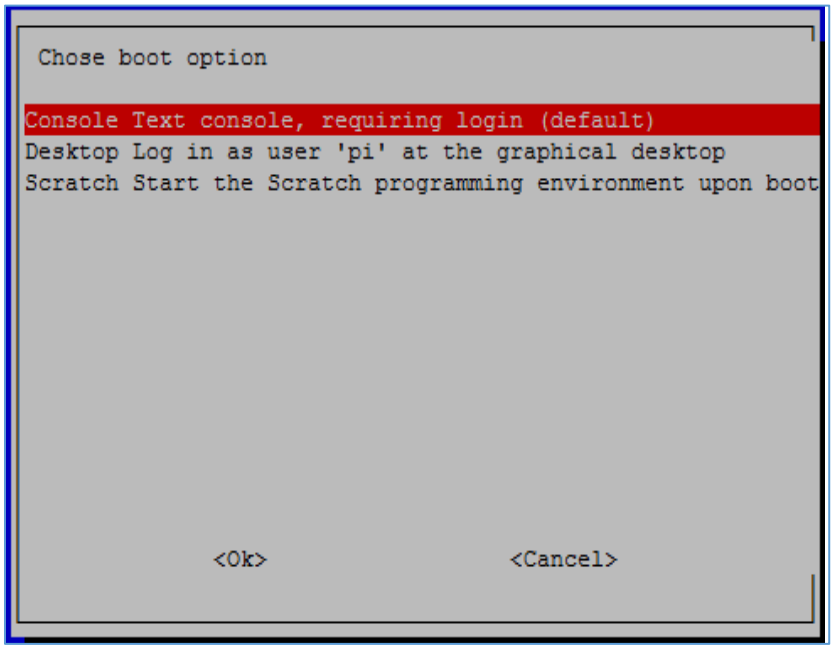


Figura 23: configuración de idioma y el teclado

Accedemos a la opción 8 en la que se encuentran las opciones avanzadas.

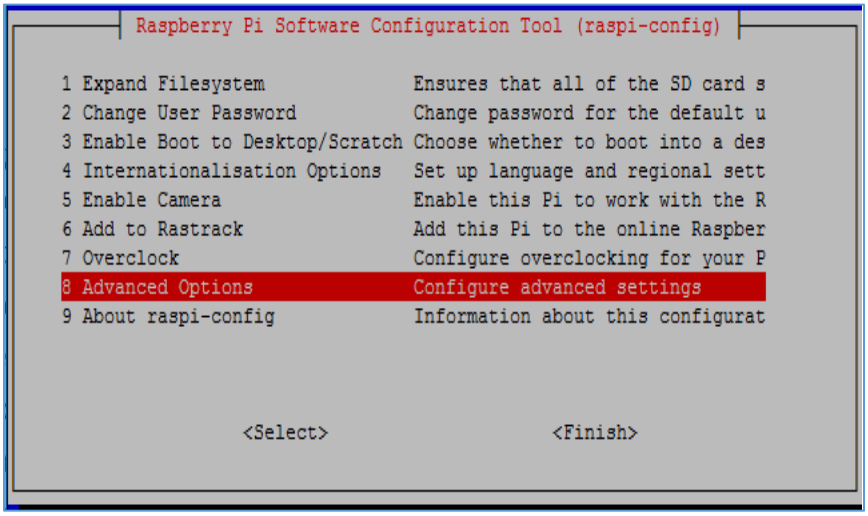


Figura 24: Interfaz de configuración de Raspberry

Dentro hay otro submenú, escogemos la opción A4 para habilitar el protocolo SSH que permitirá un posterior acceso a la Raspberry desde un ordenador vía LAN.

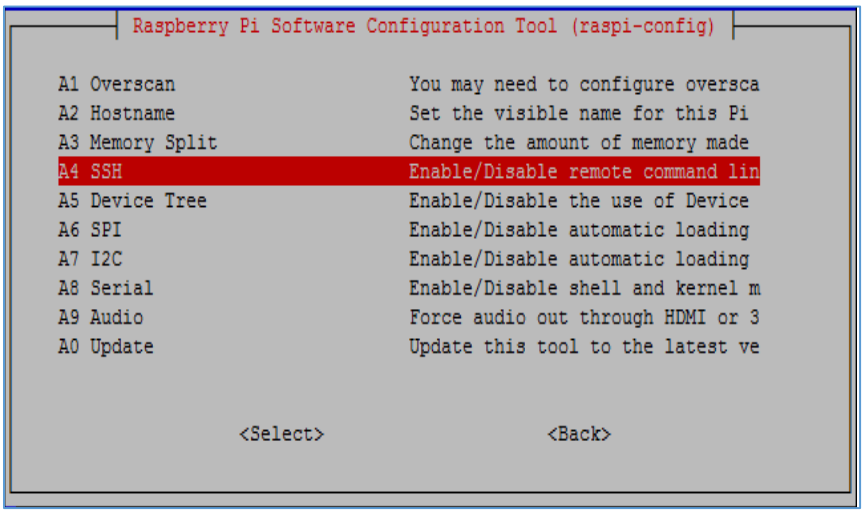


Figura 25: Interfaz de configuración de Raspberry.

Marcamos como “Enable” y presionamos ENTER.

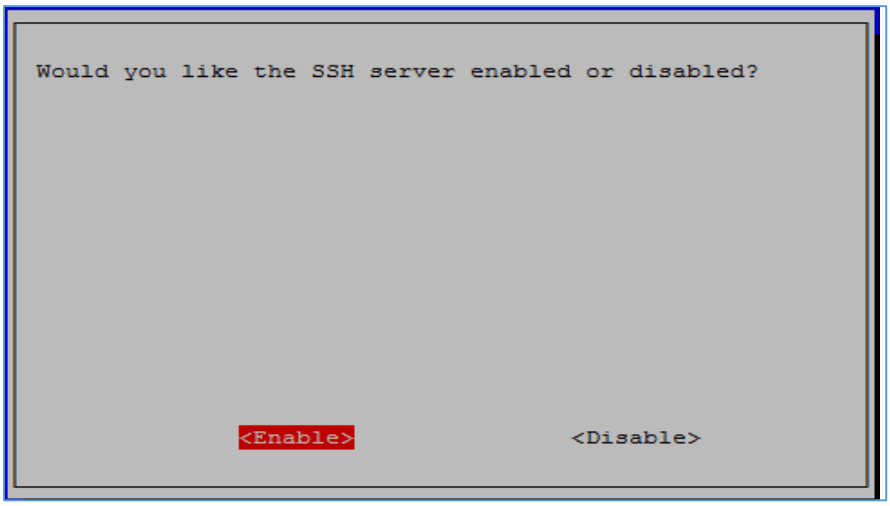


Figura 26: Interfaz de habilitación

4.2.10 Configuración básica de la red

El primer paso para configurar la red, consiste en definir las interfaces de red de este nodo, así como el nombre del nodo y la dirección del servidor de nombres (para poder acceder a contenidos de Internet). Para ello accederemos al archivo /etc/network/interfaces y los configuraremos de la siguiente manera:

source-directory /etc/network/interfaces.d

auto lo

iface lo inet loopback

 # Configuración de la red ethernet

auto eth0

iface eth0 inet static

 # Los servidores de nombres (normalmente los asigna el Router)

dns-nameservers 46.39.192.4

dns-nameservers 8.8.8.8

 # La dirección de la red ethernet

address 192.168.14.12

 # La dirección del Router

gateway 192.168.12.254

netmask 255.255.255.0

 # Configuración de la red WLAN en modo ad-hoc

allow-hotplug wlan0

iface wlan0 inet static

address 192.168.1.1

netmask 255.255.255.0

 # El canal utilizado para la red Ad-hoc
wireless-channel 1
 # El nombre de la red
wireless-essid ManetLAB
 # El modo de funcionamiento
wireless-mode ad-hoc

- La configuración anterior es la configuración del Nodo Frontera, realizamos los cambios respectivos adaptando la configuración a la LAN ya sea de desarrollo o producción.
- **Ctrl + O //Guardamos los cambios**
- **Ctrl + X //salir**
- También lo cambiamos en /etc/hosts (el nombre asociado a la IP local 127.0.1.1). Seguidamente cambiamos el nombre del host en **/etc/hostname** Le pondremos el nombre Rasp1.

4.2.11 Reinicio y actualización

Seguidamente reiniciamos el sistema mediante el mandato: **sudo shutdown –h now** para que adquiera los cambios anteriores.

Comprobaremos que el sistema accede a Internet.

Deberemos ahora de actualizar el firmware con el mandato: **sudo rpi-update** y seguidamente actualizamos todos los paquetes de software mediante los mandatos:

sudo apt-get update
sudo apt-get upgrade

4.2.12 Protocolos de encaminamiento

La búsqueda de un protocolo de encaminamiento en redes ad-hoc se ha convertido en un importante desafío debido a su complejidad. A diferencia de las redes clásicas cableadas que presuponen que la topología de la red es poco cambiante, en una Manet se utiliza como medio de transmisión el aire y está en constante cambio por la movilidad de sus nodos. Por ello no se puede utilizar los algoritmos ya existentes y se deben buscar nuevos que soporten estas condiciones, teniendo en cuenta las limitaciones del ancho de banda, la memoria reducida y la saturación por el denso tráfico que han de soportar.

4.2.13 Tipos de protocolos de encaminamiento

Para tener un encaminamiento eficiente el algoritmo debería de tener las siguientes características: Señalización mínima, mínimo tiempo de procesado, que no se produzcan bucles, que sea distribuido, que soporte la topología dinámica; que soporte enlaces unidireccionales y el modo sleep cuando el nodo está inactivo.

Se pueden diferenciar tres grandes grupos:

- Protocolos Proactivos
- Protocolos Reactivos
- Protocolos Híbridos

En este caso nos centraremos en protocolos proactivos, y que además nos permite hacer operaciones necesarias para llevar a cabo este proyecto fin de grado.

4.2.13.1 Protocolos Proactivos

Este tipo de algoritmos trata de mantener la información necesaria para el encaminamiento continuamente actualizado mediante el uso de tablas. Cada nodo tiene una o varias tablas en las que guarda la ruta que debe utilizar para llegar a cualquier nodo de la red.

Cuando la topología sufre una modificación (un nodo se incorpora, deja de formar parte o cambia de posición) se inunda la red de mensajes en modo broadcast para actualizar las rutas de todas las tablas.

Estudiaremos a fondo el protocolo OLSRD (Optimized Link State Routing Deamon) que utilizaremos como ejemplo de los modelos proactivos en las simulaciones.

4.2.13.1.1 Protocolo OLSRD

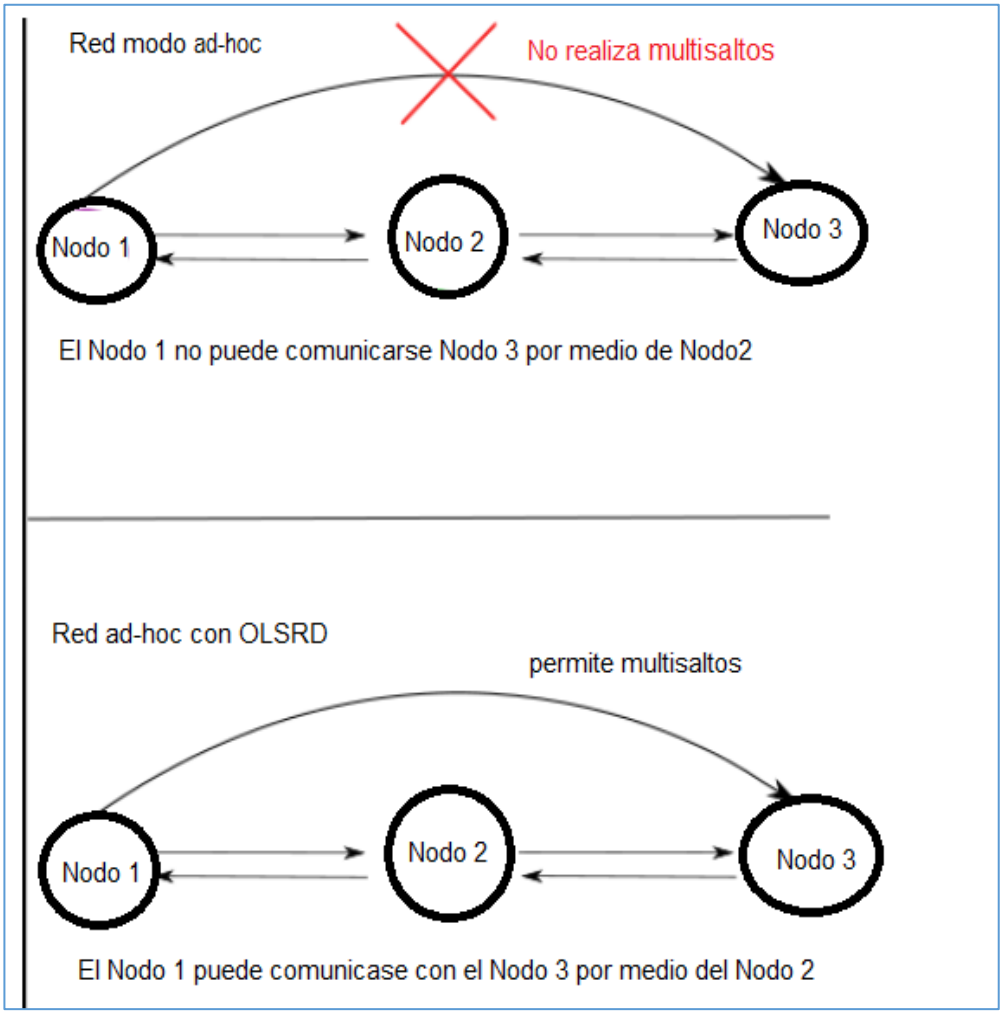


Figura 27: Muestra de posibilidad de comunicación ampliada con OLSRD.

OLSRD: Optimized Link State Routing Deamon o Demonio de Enrutamiento de Estado de Enlace, es un protocolo diseñado para trabajar con redes ad-hoc que también son conocidas como MANETs (Mobile ad hoc network).Se trata de un protocolo proactivo, es decir que actualiza la información de la topología de red con otros Routers regularmente. El protocolo busca identificar todos los destinos de la red, identificar un subconjunto suficiente de enlaces en la red, con el fin de que las rutas más cortas a todos los destinos disponibles se puedan calcular y proporcionar un conjunto de enrutamiento que contenga las rutas más cortas desde un Router a todos los destinos.

Como hemos visto, el intercambio de tantos paquetes, congestiona la red y supone un grave problema en las comunicaciones. Para solucionar esto, OLSRD utiliza la técnica de MPR (Multi Point Relay). Gracias a esta técnica se reduce el número de retransmisiones. Veamos en qué consiste esta técnica.

MPR (Multi Point Relay): La técnica MPR consiste en seleccionar un mínimo conjunto de nodos vecinos a un salto de distancia, que sean capaces de llegar a todos los nodos vecinos que se encuentran a dos saltos de distancia.

De esta forma, un nodo selecciona su conjunto de nodos MPR, y sólo puede intercambiar mensajes de control con ellos. Así se evita el enviar de forma masiva mensajes de broadcast.

Para confeccionar la lista, cada nodo utiliza el mensaje “HELLO” que envía a todos los nodos vecinos. Este paquete tiene un campo conocido como tiempo de vida (TTL, Time To Live), que es de valor 1. Al tener el TTL un valor de 1, el mensaje sólo llega a los nodos que se encuentran a un salto de distancia y no es retransmitido por la red. De esta manera cada nodo puede conocer a sus nodos vecinos y a los vecinos de estos. De esta forma se puede saber que nodos conviene seleccionar como conjunto MPR.

En la figura 28 podemos ver como se selecciona un conjunto de nodos, MPR:

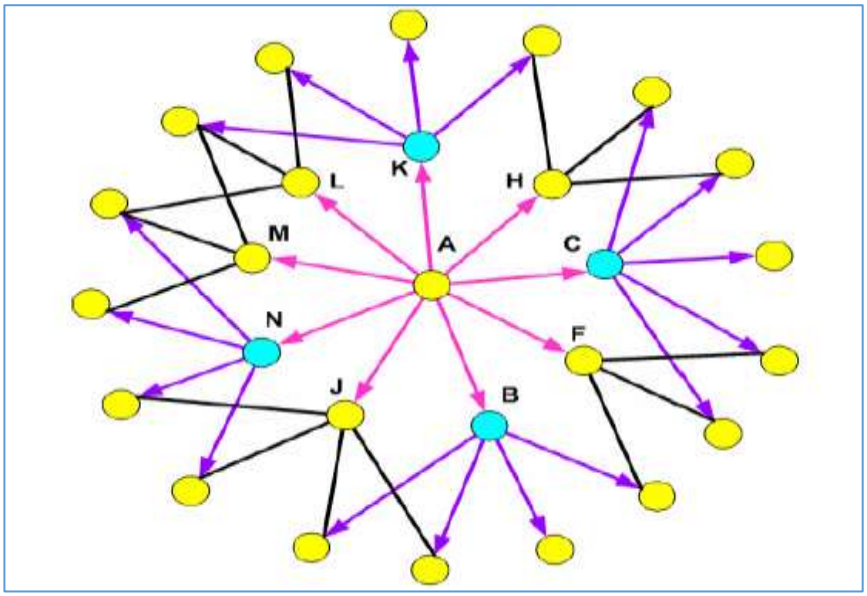


Figura 28: Selección del Multi Point Relay.

Vemos en la figura 28 como el nodo central (nodo A) selecciona el mínimo número de nodos a un salto de distancia (nodos B, C, K, N), capaces de llegar a todos los nodos que se encuentran a dos saltos de distancia. En el ejemplo el nodo A selecciona su lista de nodos MPR con los cuales sólo se enviará información evitando así la inundación de mensajes por toda la red.

4.2.14 Instalación de OLSRD

Instalamos el protocolo con:

sudo apt-get install olsrd

Con estos pasos, la configuración básica de los nodos está terminada. Detenemos el sistema y copiamos la tarjeta micro SD.

4.2.15 Configuración del Nodo 1

Volvemos a poner la tarjeta en el Nodo 1 y vamos a configurar en este paso el protocolo OLSRD. Teniendo en cuenta que este nodo será el encargado de conectar con Internet a través de un Router y permitirá, a su vez, a todos los demás nodos de la red el acceso a través de él. Dado que hay una sola dirección IP (normalmente), habremos de configurar el enmascaramiento de direcciones para que el Nodo 1 pueda dar servicio al resto de la red ad-hoc.

Se realiza a través del archivo **olsrd.conf** que se encuentra en el directorio **/etc/olsrd**. Vemos que ya está disponible en el directorio; sólo es necesario editar algunos de sus parámetros.

Ahora es necesario configurar el arranque automático del protocolo. Para ello, creamos un archivo que llamaremos **/etc/init.d/inicio_olsrd** y que contendrá lo siguiente:

#! /bin/sh
 # /etc/init.d/detector-init
 # Provides: detector-init
 # Required-Start: $all
 # Required-Stop: $remote_fs $syslog
 # Default-Start: 2 3 4 5
 # Default-Stop: 0 1 6
 # Short-Description: Script de ejemplo de arranque automático
 # Description: Script para arrancar el detector de
presencia
END INIT INFO
sudo /usr/sbin/olsrd

Ahora deberemos de definir que este archivo es ejecutable, lo que haremos con el mandato:

sudo chmod 755 /etc/init.d/inicio_olsrd

Por último, lo activamos con:

sudo update-rc.d inicio_olsrd defaults

Para probar que funciona, reiniciamos el sistema y ejecutamos el mandato:

sudo olsrd

Y nos da una respuesta tal que esta:

***** olsr.org - 0.6.6.2-git_0000000-**

hash_5766aabdb0b373cac97adf6c44c91f32 ***

Build date: 2014-06-14 06:27:56 on bm-wb-04

4.2.16 Configuración del enmascaramiento de la red

Permite, como se ha explicado, que los demás nodos de la red ad-hoc accedan a Internet. Para ello, en primer lugar habilitamos el IP forwarding:

- **sudo su**
- **sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE**

Seguidamente configuramos IPTables

- **sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE**
- **/sbin/iptables -A FORWARD -i eth0 -o wlan0 -m state –state RELATED,ESTABLISHED -j ACCEPT**
- **/sbin/iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT**

Naturalmente, tenemos que configurar el sistema de modo que el proceso de forwarding se inicie durante el periodo de boot, y para ello, quitamos el comentario (#) a la línea que dice **net.ipv4.ip_forward=1** en el archivo **/etc/sysctl.conf**

- **Ctrl + O //Guardamos los cambios**
- **Ctrl + X //salir**

Seguidamente, hemos de configurar el sistema para que IPTables tome los anteriores valores configurados durante el proceso de boot con el mandato:

sudo bash -c 'iptables-save > /etc/network/iptables'

Y para que el proceso funcione, hemos de indicar al sistema que active IPTables antes que la propia red, para ello, añadimos la línea:

pre-up iptables-restore < /etc/network/iptables

Al final del archivo **sudo nano /etc/network/interfaces**

4.2.17 Instalación del protocolo OLSRD en todos los nodos

Para configurar todo el resto de nodos de la red, el procedimiento es parecido al anterior, sólo que mucho más sencillo, solo que en este caso no se realiza las modificaciones en IPTables ni configuramos olsrd para que actúe como gateway.

Por lo tanto utilizamos la copia en micro SD que creamos para el Nodo1. Obviamente, mientras configuramos el resto de nodos, evitamos que el Nodo 1 no se encuentre activo porque tendríamos conflictos con los nombres y las direcciones IP en tanto en cuanto no se complete la configuración de cada nodo.

El contenido del archivo interfaces será ahora:

source-directory /etc/network/interfaces.d
auto lo
iface lo inet loopback
auto eth0
 # Configuración ethernet
iface eth0 inet static
 # Configuración de los servidores de nombres. Notemos que en el
Nodo # 1 no los ponemos porque normalmente el router los asigna
 # automáticamente
dns-nameservers 46.39.192.4
dns-nameservers 8.8.8.8
 # La dirección ethernet
address 192.168.2.1
netmask 255.255.255.0
 # Configuración de la red WLAN
allow-hotplug wlan0
iface wlan0 inet static
dns-nameservers 46.39.192.4
dns-nameservers 8.8.8.8
address 192.168.1.2
netmask 255.255.255.0
wireless-channel 1
wireless-essid ManetLAB
wireless-mode ad-hoc

- **Ctrl + O //Guardamos los cambios**
- **Ctrl + X //salir**

Naturalmente, deberemos de modificar los archivos **/etc/hostname** y **/etc/hosts** para que reflejen el nombre de cada uno de estos nodos (Rasp2, Rasp3, etc.).

También debemos de configurar el protocolo OLSRD exactamente igual que en el Nodo1, sólo que no se necesita el añadido de configuración para Gateway en el archivo **/etc/olsrd/olsrd.conf.**

Es necesario realizar las tareas para configuración de arranque automático del protocolo, etc. De la misma manera que en el Nodo 1.

No es necesario activar IP forwarding ni IPTables.

4.2.18 Pruebas de las tablas de rutas

Después de configurados (y reiniciados los nodos), podemos probar que las tablas de rutas de cada uno de los nodos se establecen de manera automática, para lo que resulta útil el mandato **route**, que nos daría una salida como la siguiente (ejemplo con el nodo1, nodo2 y nodo3 conectados simultáneamente).

```
root@rasp1:/home/pi# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.12.254  0.0.0.0         UG    0      0        0 eth0
default         192.168.12.254  0.0.0.0         UG    202    0        0 eth0
link-local      *               255.255.0.0     U     303    0        0 wlan0
192.168.1.0     *               255.255.255.0   U     0      0        0 wlan0
192.168.3.0     192.168.1.3     255.255.255.0   UG    0      0        0 wlan0
192.168.12.0    *               255.255.255.0   U     202    0        0 eth0
192.168.14.0    *               255.255.255.0   U     0      0        0 eth0
root@rasp1:/home/pi# ip route add 192.168.2.0/24 via 192.168.1.2
root@rasp1:/home/pi# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.12.254  0.0.0.0         UG    0      0        0 eth0
default         192.168.12.254  0.0.0.0         UG    202    0        0 eth0
link-local      *               255.255.0.0     U     303    0        0 wlan0
192.168.1.0     *               255.255.255.0   U     0      0        0 wlan0
192.168.2.0     192.168.1.2     255.255.255.0   UG    0      0        0 wlan0
192.168.3.0     192.168.1.3     255.255.255.0   UG    0      0        0 wlan0
192.168.12.0    *               255.255.255.0   U     202    0        0 eth0
192.168.14.0    *               255.255.255.0   U     0      0        0 eth0
root@rasp1:/home/pi#
```

Figura 29: Tabla de rutas cuando todos los nodos son directamente visibles entre sí.

En este ejemplo, estamos conectados al nodo 1 (Rasp1) que tiene una IP ethernet de192.168.12.20 y una IP WLAN de 192.168.1.1.

Podemos apreciar que el gateway por defecto (para acceder al exterior de la red ManetLAB) es el 192.168.168.254 que es la IP WLAN del Nodo 1.

Las líneas 3 y 4 nos dicen cómo alcanzar las interfaces ethernet del resto de los demás nodos.

Como en este caso, todos los nodos están directamente accesibles entre sí, naturalmente, el modo de alcanzarlas es a través de sus propias interfaces WLAN.

Las dos últimas líneas nos indican cómo alcanzar las interfaces WLAN del resto de nodos.

Como en el caso anterior, al ser todas visibles entre sí, no existen saltos en la red y se accede directamente a cada una de ellas.

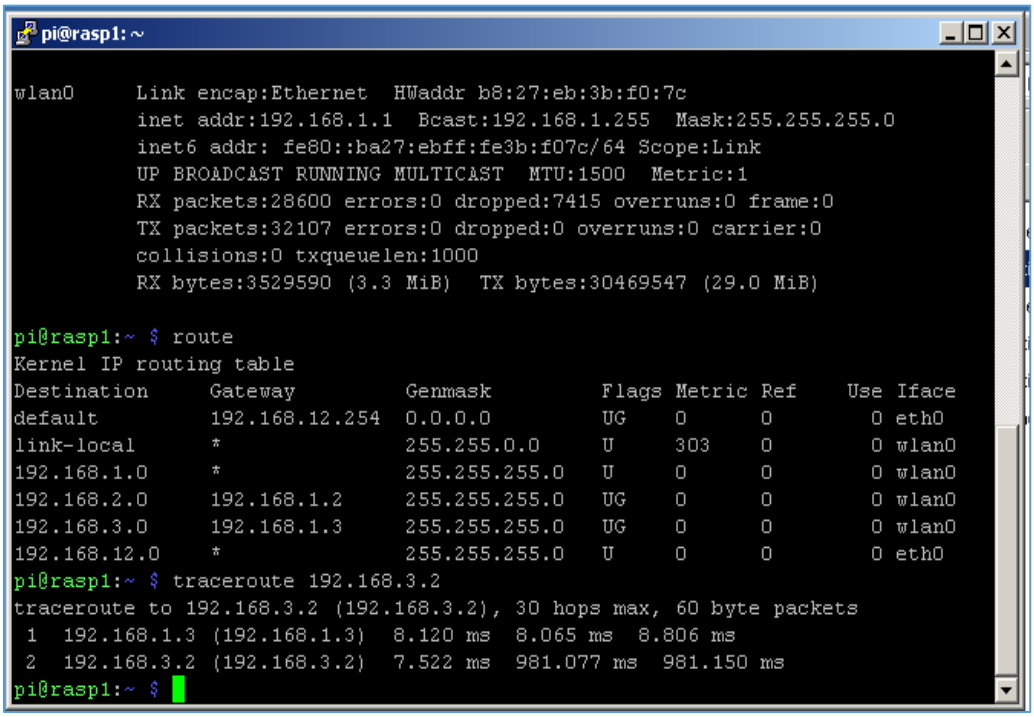


Figura 30: Tabla de rutas para un caso de comunicación multisalto

En este segundo ejemplo, el Nodo 3 se encuentra muy alejado del Nodo 2 y en medio está el Nodo 1. Si nos conectamos al Nodo 1, podemos comprobar en la tabla de rutas que para alcanzar al siguiente nodo de la red, solamente hay dos saltos.

4.2.19 Configuración manual de dirección IP para dispositivos finales

Para que un dispositivo final se comunique a través de la red, se debe configurar con la información de dirección IP correcta. De modo similar al de una SVI, el dispositivo final se debe configurar con una dirección IP y una máscara de subred. Esta información se configura en los parámetros de la PC.

Para que un dispositivo final se conecte correctamente a la red, se deben configurar todos estos parámetros. Esta información se configura en los parámetros de red de la PC. Además de la información de dirección IP y máscara de subred, es posible configurar la información del gateway predeterminado y del servidor DNS, como se muestra en la Figura 31.

La dirección de gateway predeterminado es la dirección IP de la interfaz del Nodo 1 que se utiliza para que el tráfico de la red salga de la red local. El gateway predeterminado es una dirección IP que, por lo general, asigna el administrador de red y se utiliza cuando se debe en rutar tráfico a otra red.

La dirección del servidor DNS es la dirección IP del servidor del Sistema de nombres de dominios (DNS, Domain Name System), que se utiliza para traducir direcciones IP a direcciones Web, como www.marca.com. A todos los dispositivos de Internet se les asigna una dirección IP, mediante la cual se accede a ellos. Sin embargo, resulta más fácil recordar nombres que números. Por lo tanto, los sitios Web tienen nombres para simplificar el proceso. El servidor DNS se utiliza para mantener la asignación entre las direcciones IP y los nombres de los diversos dispositivos.

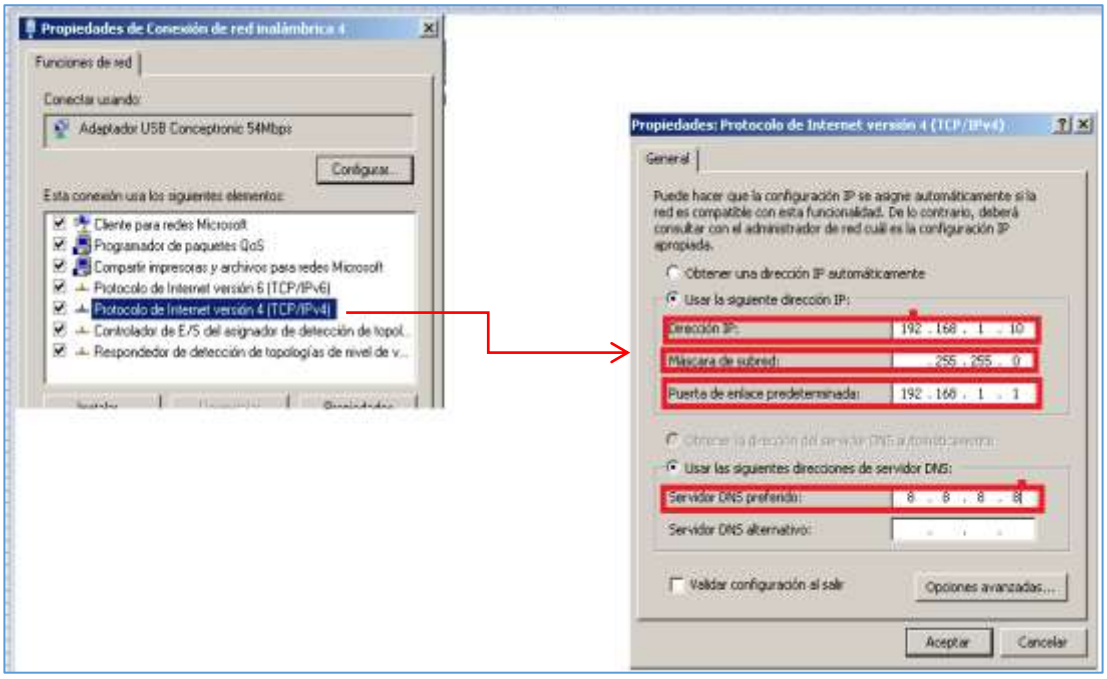


Figura 31: Direccionamiento de dispositivos finales

4.2.20 Configuración automática de direcciones IP para dispositivos finales

La información de dirección IP se puede introducir en la PC en forma manual o mediante el Protocolo de configuración dinámica de host (DHCP). El protocolo DHCP permite configurar la información de IP de los dispositivos finales de manera automática.

DHCP es una tecnología que se utiliza en casi todas las redes comerciales. Para comprender mejor por qué DHCP es tan popular, tenga en cuenta todo el trabajo adicional que habría que realizar sin este protocolo.

DHCP permite la configuración automática de direcciones IPv4 para cada dispositivo final de una red con DHCP habilitado. Suponemos la cantidad de tiempo que le llevaría si cada vez que se conectara a la red tuviera que introducir manualmente la dirección IP, la máscara de subred, el gateway predeterminado y el servidor DNS. Si multiplicamos eso por cada usuario y cada uno de los dispositivos en la red, eso tardaría mucho tiempo y costaría muchísimo trabajo.

DHCP es un ejemplo de la tecnología en su máxima expresión. Uno de los propósitos principales de cualquier tecnología es facilitar las tareas que se desean o se deben realizar. Con DHCP, el usuario final ingresa al área que abarca una red determinada, conecta un cable Ethernet o habilita una conexión inalámbrica e, inmediatamente, se le asigna la información de IPv4 necesaria para comunicarse de manera correcta a través de la red.

Para configurar el protocolo DHCP en un equipo Debian, solo tenemos que seguir los siguientes pasos:

- Los mandatos para la instalación de DHCP son:
  sudo apt-get install isc-dhcp-server
- Configuración de ISC-DHCP-SERVER. Toda la configuración se realiza en el archivo dhcpd.conf ubicado en /etc/dhcp/. En este archivo comentamos o borramos las siguientes líneas:
  option domain-name "example.org";
  option domain-name-servers ns1.example.org, ns2.example.org;

En la sección de abajo descomentamos authoritative:

#if this DHCP server is the official DHCP server for the local
#network, the authoritative directive should be uncommented.

authoritative;

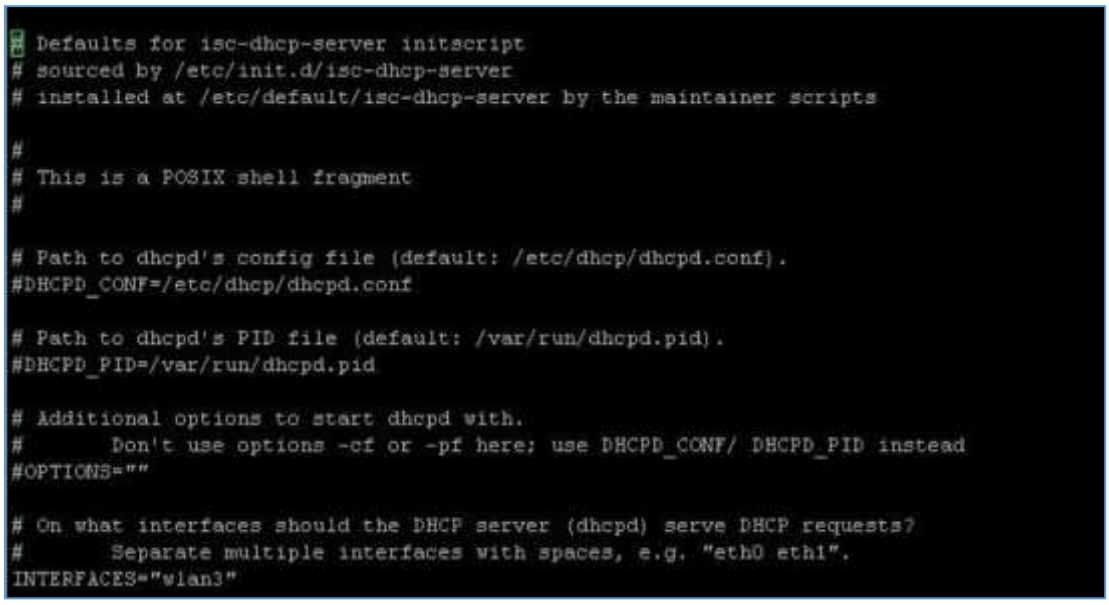
Ahora definimos la red y el rango de redes que el servidor proporcionara a las peticiones DHCP para ello escribimos lo siguiente al final de este archivo:

subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.10 192.168.1.50;
option broadcast-address 192.168.1.255;
option routers 192.168.1.1;
default-lease-time 600;
max-lease-time 7200;
option domain-name "local-network";
option domain-name-servers 8.8.8.8, 8.8.4.4;
}

Con lo anterior se le está configurando de tal manera que todo aquel dispositivo que solicite DHCP al servidor se le otorgara una IP dentro del rango de 192.168.1.10 y 192.168.1.50, dando un máximo de hasta 40 direcciones IPs, pertenecientes a la red 192.168.1.0 y que usara los servidores DNS de Google el 8.8.8.8 o el 8.8.4.4. Una vez realizadas todas las modificaciones a los archivos guardamos con el comando **Ctrl + O** y salimos del editor nano con **Ctrl + X**.

Ahora Configuramos la interfaz por la cual el servidor escuchara las peticiones DHCP, para esto, editamos el archivo isc-dhcp-server ubicado en /etc/default/

Buscamos la línea “INTERFACES” y colocamos el nombre de la interfaz por la cual escuchara las peticiones, en todo caso será la otra interfaz de red inalámbrica que posea, recordar que por una interfaz se conectara a la red ManetLAB y por la otra dará el servicio AP, es por la interfaz que brindara el servicio AP que escuchara las peticiones. Para este tutorial ocupamos la interfaz Wlan0 en el nodo 3, para obtener el identificador de su interfaz de red.



```
# Defaults for isc-dhcp-server initscript
# sourced by /etc/init.d/isc-dhcp-server
# installed at /etc/default/isc-dhcp-server by the maintainer scripts

#
# This is a POSIX shell fragment
#

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPD_CONF=/etc/dhcp/dhcpd.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPD_PID=/var/run/dhcpd.pid

# Additional options to start dhcpd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="wlan3"
```

Figura 32: Archivo de configuración por default de isc-dhcp-server.

4.2.21 Prueba de la dirección de bucle invertido en un dispositivo final

Prueba de loopback

El comando ping se utiliza para verificar la configuración IP interna en un host local. Esta prueba se realiza utilizando el comando ping en una dirección reservada denominada “dirección de loopback” (127.0.0.1). El protocolo TCP/IP define la dirección de loopback, 127.0.0.1, como una dirección reservada que permite enrutar los paquetes de regreso al host.

Los comandos ping se introducen en una línea de comandos en el host local con la siguiente sintaxis:

```
pi@rasp1:~ $ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.236 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.127 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.071 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.133 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.138 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.080 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.143 ms
```

64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.128 ms
64 bytes from 127.0.0.1: icmp_seq=9 ttl=64 time=0.076 ms
64 bytes from 127.0.0.1: icmp_seq=10 ttl=64 time=0.145 ms
^C
--- 127.0.0.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8994ms
rtt min/avg/max/mdev = 0.071/0.127/0.236/0.047 ms

El resultado indica que se enviaron ocho paquetes de prueba de 64 bytes cada uno desde el host 127.0.0.1 y se devolvieron a este en un tiempo de menos de 1 ms. Esta solicitud de ping correcta verifica que la tarjeta de interfaz de red, los controladores y la implementación del protocolo TCP/IP funcionan correctamente.

4.2.22 Prueba de la conectividad de extremo a extremo

Prueba de la conectividad de PC a Raspberry PI 3B

El comando ping se puede utilizar en una PC de la misma forma que en un dispositivo intermediario, nuestro caso Raspberry PI 3B. En la Figura 33, se muestra que el ping de la PC a la dirección IP de la interfaz ManetLAB del Raspberry 1, 192.168.12.254, debe ser correcto.

Prueba de la conectividad de extremo a extremo.

La dirección IP de la PC es 192.168.1.10, con una máscara de subred 255.255.255.0 y un gateway predeterminado 192.168.1.1.

La dirección IP de Raspberry 1 es 192.168.1.1, con una máscara de subred 255.255.255.0 y un gateway predeterminado 192.168.12.254.

El ping de la PC1 a la Raspberry 1 también debe ser correcto. Si un ping de la PC a la Raspberry 1 se realiza correctamente, se verifica la conectividad de extremo a extremo en la red.

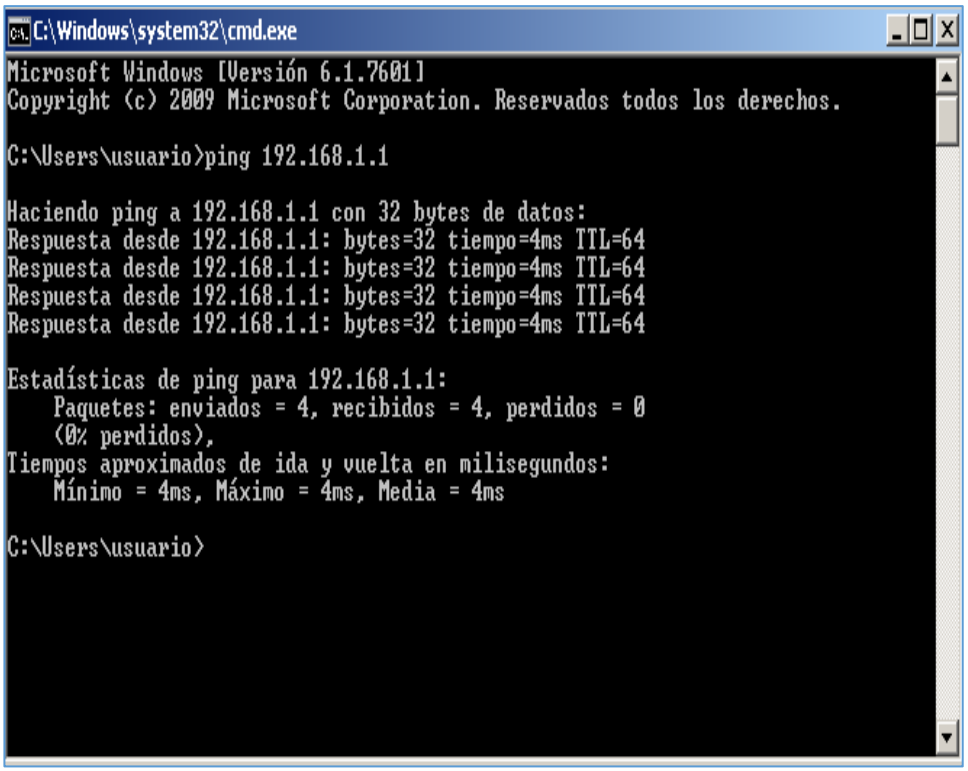


Figura 33: prueba de la conectividad de extremo a extremo

4.2.23 Traceroute, prueba de ruta

Ping se utiliza para probar la conectividad entre dos hosts, pero no proporciona información sobre los detalles de los dispositivos entre los hosts. Traceroute (tracert) es una utilidad que genera una lista de saltos que se alcanzaron correctamente a lo largo de la ruta. Esta lista puede proporcionar información importante sobre la verificación y la resolución de problemas. Si los datos llegan al destino, el rastreo indica la interfaz de cada router o Raspberry que aparece en la ruta entre los hosts. Si los datos fallan en algún salto a lo largo del camino, la dirección del último router que respondió al rastreo puede indicar dónde se encuentra el problema o las restricciones de seguridad.

Tiempo de ida y vuelta (RTT)

El uso de traceroute proporciona el tiempo de ida y vuelta para cada salto a lo largo de la ruta e indica si se produce una falla en la respuesta del salto. El tiempo de ida y vuelta es el tiempo que le lleva a un paquete llegar al host remoto y el tiempo que la respuesta del host demora en regresar. Se utiliza un asterisco (*) para indicar un paquete perdido o sin respuesta.

Esta información puede ser utilizada para ubicar un router problemático en el camino. Si en la pantalla se muestran tiempos de respuesta elevados o pérdidas de datos de un salto particular, esto constituye un indicio de que los recursos del router o sus conexiones pueden estar sobrecargados.

Tiempo de vida (TTL) de IPv4 y Límite de saltos de IPv6

Traceroute utiliza una función del campo TTL en IPv4 y del campo Límite de saltos en IPv6 en los encabezados de capa 3, junto con el mensaje de tiempo superado de ICMP.

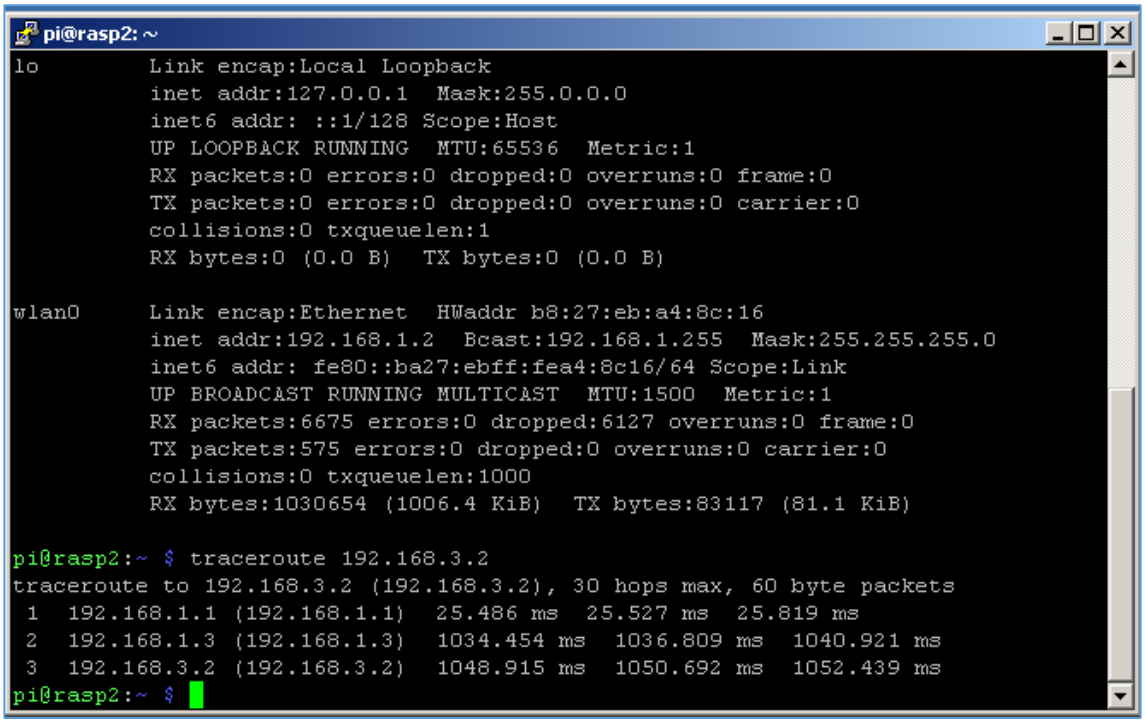


Figura 34: Traceroute (tracert).Prueba de la rutas-comunicación multisalto

4.3 CONFIGURACIÓN DE SERVICIOS BLUETOOTH Y SOFTWARE NODEJS EN MINIORDENADORES RASPBERRY PI 3 B

4.3.1 Descripción

Deseamos configurar en un nodo Raspberry el software base que nos permitirá desarrollar aplicaciones basadas en Bluetooth (en un caso sencillo, recibir la ráfaga de datos procedentes de un iBeacon) y que además permita enviar los datos recibidos de los dispositivos Bluetooth a un sistema de proceso central (que puede estar en otro nodo Raspberry o bien puede ser un ordenador convencional). Para ello utilizaremos la tecnología denominada “websocket” que será controlada, además de las comunicaciones Bluetooth mediante el software de desarrollo Nodejs, que no es más que un framework que nos permitirá desarrollar aplicaciones Javascript fuera de un navegador.

4.3.2 Instalación paquetes

El software básico para control de Bluetooth ya se encuentra instalado por defecto en el sistema operativo Raspbian. Instalaremos los siguientes paquetes adicionales:

NetTalk: Permite que un ordenador equipado con sistema operativo macOS pueda acceder a archivos dentro de un Raspberry como si se tratase de una unidad remota de red.

Nodejs versión 6: El framework de desarrollo para Javascript.

4.3.2.1 Instalación de nettalk

El proceso es sencillo, basta con ejecutar los siguientes mandatos:

sudo apt-get update

sudo apt-get install netatalk

Una vez instalado, hemos de configurar aquellos directorios a los que deseamos tener acceso mediante macOS.

Detenemos el servicio:

sudo /etc/init.d/netatalk stop

Editamos el archivo de configuración AppleVolumes.default:

sudo nano /etc/netatalk/AppleVolumes.default

Buscamos (hacia el final del archivo) la línea que contiene:

~/ "Home Directory"

Ahora podemos añadir directorios adicionales (la línea ~/ se refiere al directorio raíz del usuario que se conecte. Podemos asignarle cualquier nombre, en este caso es

“Home Directory”. Este nombre aparecerá como nombre de la carpeta en macOS. Ya está por defecto, pero podemos añadir nuevos directorios, ejemplo:

~/ “Home Directory"

/media “Media"

/mnt "Mount"

Justo antes de esta línea, en la que empieza por: DEFAULT, añadimos rw para que podamos leer y escribir en los directorios, con lo que la línea final queda como:

:DEFAULT: options:upriv,usedots,rw

Activamos de nuevo el servicio mediante:

sudo /etc/init.d/netatalk start

4.3.2.2 Instalación de nodejs

Para la instalación de paquetes Javascrit de forma local, podemos recurrir a las indicaciones suministradas en: https://docs.npmjs.com/getting-started/installing-npm-packages-locally.En nuestro caso, deseamos instalar tanto los paquetes que nos permitirán manejar el sistema Bluetooth como los websockets desde un programa escrito en Javascript. Utilizaremos para ello la distribución “npm” que contiene infinidad de paquetes interesantes.

Para ello, nos vamos a nuestro directorio raíz (/home/pi) y ejecutamos los siguientes mandatos:

npm install bleacon

npm install noble

npm install ws

Este proceso nos crea en nuestro directorio personal los directorios:

node_modules/bleacon (Gestión de dispositivos iBeacon)

node_modules/noble (Control de sistemas Bluetooth LTE)

node_modules/ws (Gestión de Websockets)

4.3.3 Instalación de los ejecutables nodejs y de los certificados digitales.

4.3.3.1 Nodo principal

En el directorio de usuario del nodo principal (/home/pi) copiamos el archivo “servidor_cifrado.js”. Seguidamente, creamos un directorio llamado “certificados” y copiamos en él los dos archivos: “cert.pem” y “key.pem”.

Para ejecutar el programa servidor, lo haremos con el mandato:

“sudo node servidor_cifrado.js”

A continuación debemos de guardar los datos procedentes de cada uno de los nodos clientes en un formato .txt para ello se ejecuta con el mandato:

“node servidor_cifrado.js>datos.txt”.

Para terminar la ejecución, lo haremos con Ctrl-C.

Finalmente, si deseamos acceder a los datos anteriormente guardados, lo haremos con el mandato:

“nano datos” ó **“cat datos”**

4.3.3.2 Nodos secundarios

Basta con copiar el archivo “nodo_cifrado.js” en el directorio /home/pi. Se ejecuta con el mandato:

“sudo nodejs cliente_cifrado.js”.

Nota importante: En el archivo “cliente_cifrado.js” (que es texto plano) está descrita la dirección IP del nodo principal, en la línea: “var ws = new WebSocket ('wss://192.168.1.1:8443', null, {rejectUnauthorized: false});”. Ponemos la dirección real del nodo principal.

Si activamos el iBeacon, veremos que en el terminal del nodo principal comienzan a aparecer las medidas de potencia realizadas en los nodos secundarios.

4.4 IMPLEMENTACIÓN DE LA APLICACIÓN Y RESULTADOS

4.4.1 Descripción de la aplicación

El objetivo principal de este proyecto consiste en procesar niveles de potencia de las señales recibidas por los iBeacons en una red MANET y, a su vez, transmitir adecuadamente y de forma cifrada esta información a un sistema central de procesado de información, utilizando como nodos de la red los equipos Raspberry Pi en concreto modelo 3B, que cumpla las especificaciones descritas en el siguiente punto.

4.4.2 Protocolos de seguridad en la capa de aplicación

El sistema permite recibir, transmitir y procesar los niveles de potencia y otros datos de interés de forma cifrada, con las funcionalidades habituales:

- **Protocolo WSS**: WS-Security (Seguridad en Servicios Web) es un protocolo de comunicaciones que suministra un medio para aplicar la seguridad de servicios web, equivalentemente al protocolo HTTPS.El protocolo contiene especificaciones sobre cómo debe garantizarse la integridad en mensajería de servicios Web. El protocolo WSS incluye detalles en el uso de formatos de certificado tales como X.509 codificado base 64.
- **Protocolo HTTP y HTTPS**: La navegación en páginas web se basa en los protocolos HTTP (Hypertext Tranfer Protocol) y HTTPS (Hypertext Tranfer Protocol Secure). Sin entrar en detalle en los protocolos, lo importante es que especifican el lenguaje en el que los navegadores (Google Chrome, Internet Explorer, Mozilla Firefox, Apple Safari,……), servidores de páginas Web (hostings) e intermediarios (routers) se entienden para que nosotros podamos ver páginas web de una manera cómoda. La diferencia entre ambos es la “S” final, ya que el **HTTPS implementa el protocolo SSL** que da seguridad a HTTP.
- **SSL ¿qué aporta?**: En HTTP la navegación tenía dos carencias que SSL en HTTPS soluciona:
- **Encriptación**: en HTTP la comunicación se realiza en lo que se denomina texto plano, es decir, sin encriptar. Por lo tanto, cualquier persona que con unos conocimientos suficientes de informática intercepte esa comunicación será capaz de ver dicha información. Esto puede ser especialmente en casos de información sensible (raza, religión, etc) así como en el envió de contraseñas, información de tarjetas de crédito, etc. HTTPS hace que toda

la información se envié encriptada. Por lo tanto, si enviamos nuestra contraseña “micontraseña” en un formulario web, con HTTP probablemente viajara como “micontrseña” que con HTTPS ira como algo del estilo “FajdfeVcsEJAsncQQi3d=”. De esta segunda forma, aunque nuestra comunicación sea interceptada, nadie conocerá nuestra contraseña.

- **Identidad**: con HTTP no tenemos la certeza de que el destinatario de la comunicación, por ejemplo una plataforma de pago, sea realmente quien dice ser y no un suplantador de identidad. Con HTTPS esto se soluciona gracias a que la conversación, como decíamos en el punto anterior va encriptada. Lo interesante de esta encriptación es que se realiza de una forma que solo el destinatario original es capaz de desencriptar correctamente la información, por lo que aunque alguien intercepte la comunicación o intente suplantar la identidad del destinatario, no será capaz de desencriptar el mensaje.
- **Certificadores**: Decíamos que HTTPS utiliza SSL y que este utiliza algoritmos de encriptación asimétrica con clave pública y privada. Pero ¿cómo sabemos que la clave pública del servidor web destino es la correcta y así aseguramos de la identidad de dicho servidor? Para eso confiamos en entidades certificadoras intermediarias, que nos dicen si aquel con el que intentamos comunicarnos es quien dice ser.

Básicamente estos certificadores comprueban que una empresa que solicita un certificado SSL es quien dice ser, y se le da un certificado que él debe guardar. Cuando un cliente inicia una comunicación con un servidor, el servidor le envía su clave pública. El cliente, una vez recibida dicha clave, le pregunta al certificador si esa clave es correcta y pertenece a quien dice que le pertenece. Si esto es así, la comunicación continúa. En otro caso, la comunicación se para pues no es considerada segura.

<u>Como generar clave PEM de nuestro certificado:</u>

Requisitos:

- Certificado digital valido.
- JAVA 7.
- Internet Explorer.

El fichero PEM es un fichero público que contiene texto (o ASCII) codificado en base 64 entre las etiquetas:

----- BEGIN CERTIFICATE ----- y ----- END CERTIFICATE -----

Pasos a seguir:

- Paso 1:

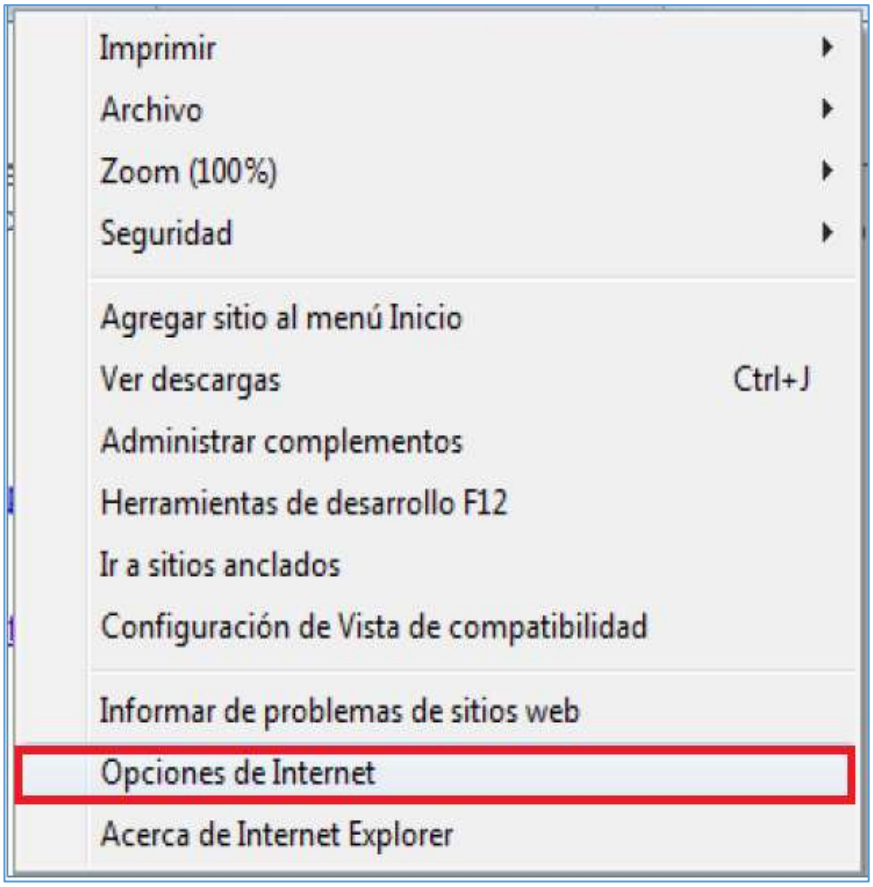


Figura 35: Menú Opciones de Internet.

- Paso 2:

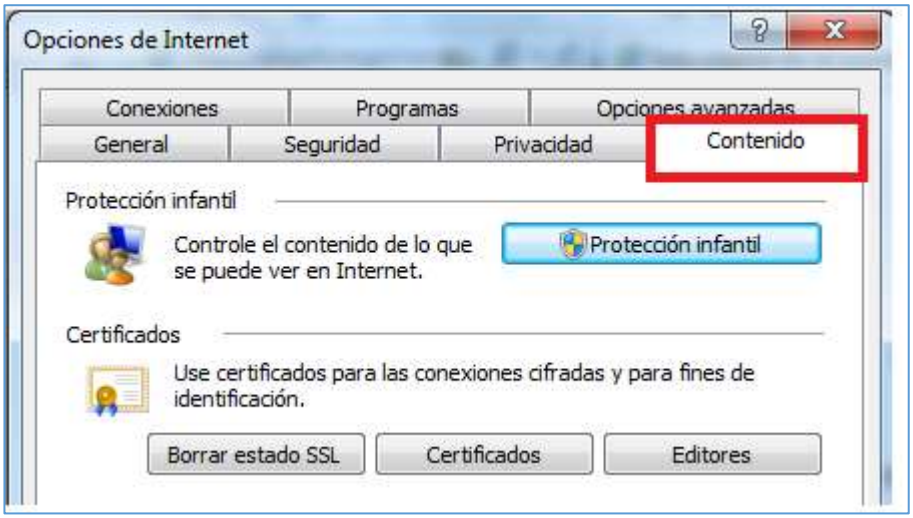


Figura 36: Pestaña contenido.

- Paso 3:

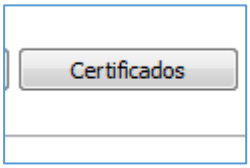


Figura 37: Botón Certificados.

- Paso 4:

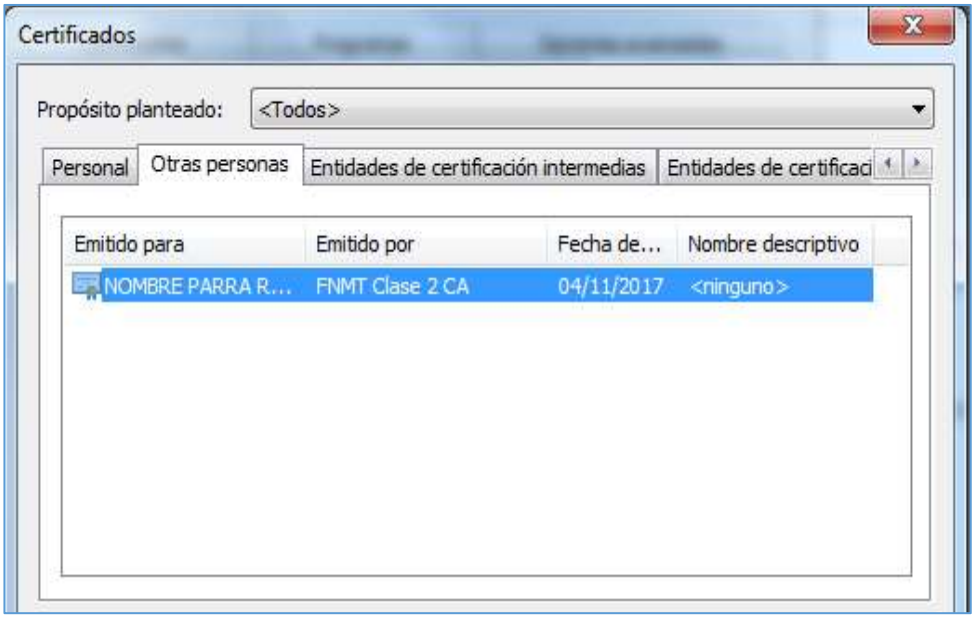


Figura 38: Seleccionar el certificado.

- Paso 5:

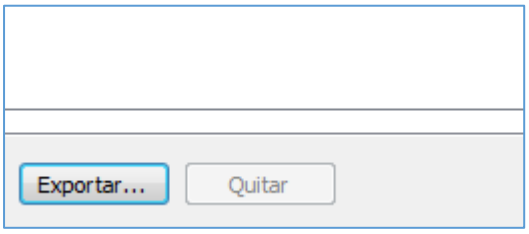


Figura 39: Pulsar exportar.

- Paso 6. Mensaje: Este es el Asistente para la exportación de certificados, pulsar SIGUIENTE.
- Paso 7. Mensaje: Elegimos la exportación de la clave con el certificado: pulsamos No exportar la clave privada.

- Paso 8: Mensaje: formato de archivo de exportación, pulsar X.509 codificado base 64 (.CER).
- Paso 9. Generamos un fichero en nuestro escritorio que se llama, por ejemplo MICERTIFICADO y pulsamos Siguiente.
- Paso 10. Mensaje: Finalización del Asistente para exportación de certificados, pulsamos FINALIZAR. Se debe abrir el BLOC de NOTAS en Todos los Programas / Accesorios / Bloc de Notas.
- Paso 11. En ARCHIVO, pulsamos ABRIR.
- Paso 12.Donde aparece el tipo de 'Documentos de texto (*.txt).

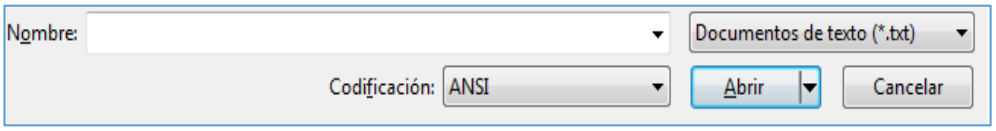


Figura 40: Pestaña Documentos de texto

Lo cambiamos por:

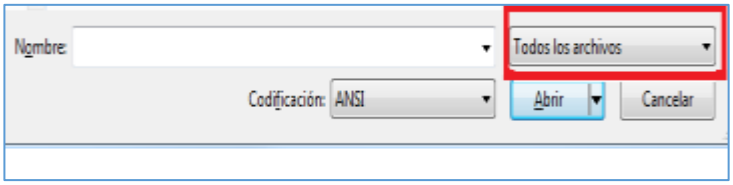


Figura 41: Pestaña Todos los documentos.

- Paso 13. Buscamos en el escritorio el certificado generado previamente.

Figura 42: Certificado generado previamente.

- Paso 14. Código/ fichero PEM:

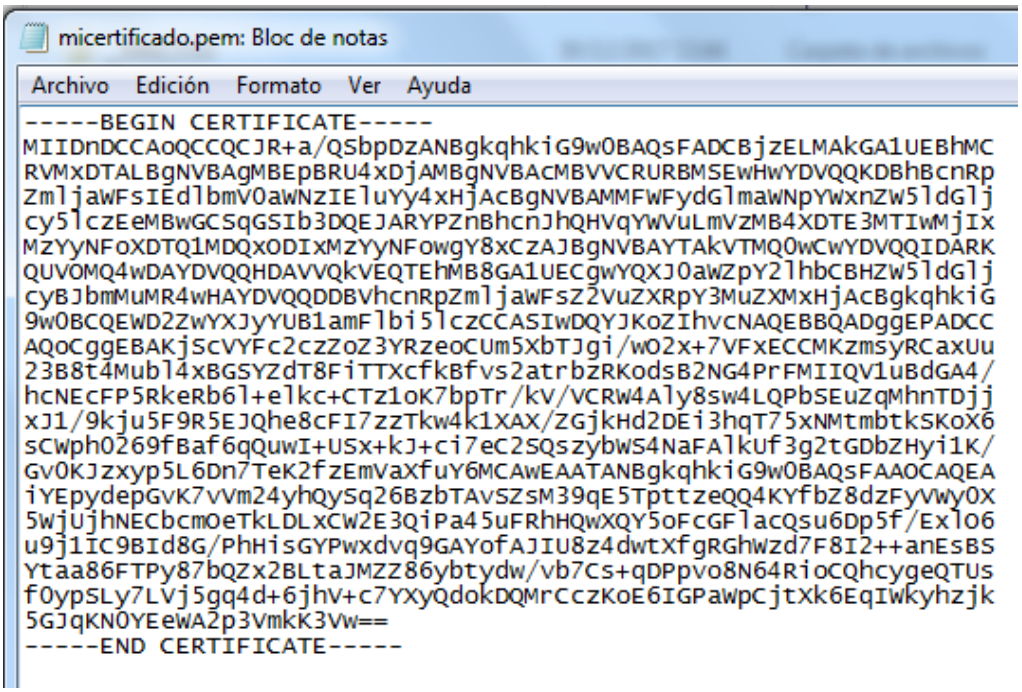


Figura 43: código fichero PEM.

Entramos en www.face.gob.es, en el campo PEM, PEGAMOS EL CONTENIDO PEM que acabamos de generar.

- Queremos recordar **Base 64** como un sistema de numeración posicional que usa 64 como base. Es la mayor potencia de dos que puede ser representada usando únicamente los caracteres imprimibles de ASCII. Esto ha proporcionado su uso para la codificación de correos electrónicos y otras aplicaciones Todas las variantes famosas que se conocen con el nombre de Base64 usan el rango de caracteres A-Z, a-z y 0-9 en este orden para los primeros 62 dígitos, pero los símbolos escogidos para los últimos dos dígitos varían considerablemente de unas a otras.
- **ASCII** (American Standard Code for Information Interchange -Código Estándar Estadounidense para el Intercambio de Información), pronunciado generalmente [áski], es un código de caracteres basado en el alfabeto latino, tal como se usa en inglés moderno.

El código ASCII utiliza 7 bits para representar los caracteres, aunque inicialmente empleaba un bit adicional (bit de paridad) que se usaba para detectar errores en la transmisión. A menudo se llama incorrectamente ASCII a varios códigos de caracteres de 8 bits que extienden el ASCII con caracteres propios de idiomas distintos al inglés, como el estándar ISO/IEC 8859-1.

- **WebSocket**: Web socket es una tecnología que proporciona un canal de comunicación bidireccional y full-duplex sobre un único socket TCP. Está diseñada para ser implementada en navegadores y servidores web, pero puede utilizarse por cualquier aplicación cliente/servidor. Debido a que las conexiones TCP comunes sobre puertos diferentes al 80 son habitualmente bloqueadas por los administradores de redes, el uso de esta tecnología proporcionaría una solución a este tipo de limitaciones proveyendo una funcionalidad similar a la apertura de varias conexiones en distintos puertos, pero multiplexando diferentes servicios WebSocket sobre un único puerto TCP (a costa de una pequeña sobrecarga del protocolo).

4.4.3 Interacción iBeacons con Raspberry Pi 3B

Hemos visto que un beacon suministra tres parámetros: UUUID, Mayor y Minor y que dichos parámetros identifican de forma unívoca a un beacon que pertenece a un grupo y, a su vez, pertenece a una empresa determinada.

De esta manera, el receptor, una vez recibidos los parámetros de beacon, invoca al servidor y lo va procesando de forma cifrada.

La interacción de beacon con Raspberry Pi 3B se puede resumir en el siguiente diagrama esquemático:

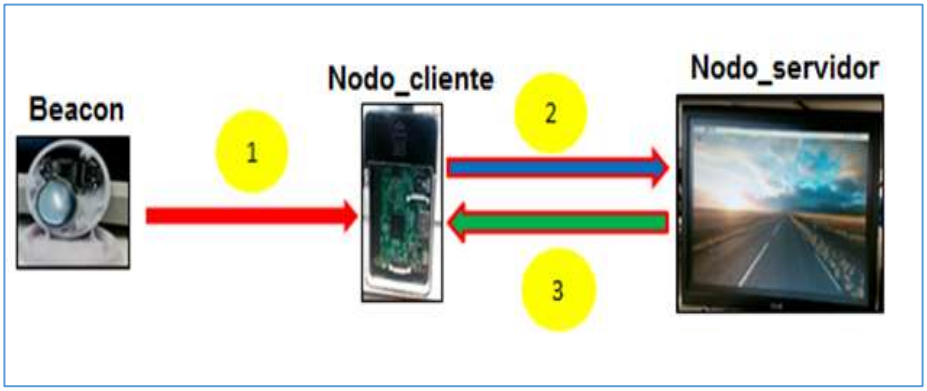


Figura 44: Interacción iBeacons con Raspberry Pi 3B.

La secuencia de acciones de la figura anterior es la siguiente:

1. El beacon emite una ráfaga de datos cada cierto tiempo con los datos UUUID, Mayor, Minor y otros datos de interés, en nuestro caso los niveles de potencias

2. El nodo cliente (Raspberry Pi 3B) invoca a servidor (otro Raspberry Pi 3B) al que lo pasa como parámetros de entrada los dados anteriores.

3. El servidor va procesando los niveles de potencias procedentes de los nodos clientes, con lo cual podemos a determinar con facilidad el nodo con mayor o menor potencia recibida y, además envía un mensaje de vuelta al cliente, es decir, en un tiempo determinado uno de los nodos clientes puede estar fuera de servicio, en otro caso el servidor tiene que seguir funcionando…

4.4.4 Interpretación del esquema de la situación real de la aplicación

Este método consiste en separar los tres Raspberry en diferentes laboratorios que posteriormente se va analizar la situación según los datos obtenidos en cada simulación, la siguiente imagen muestra la ubicación de cada Raspberry en particular:

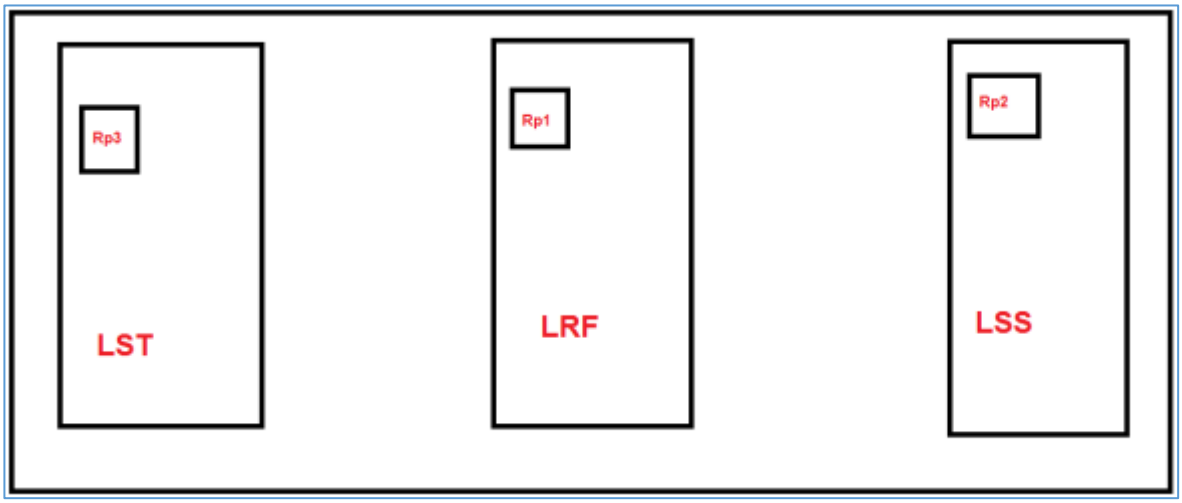


Figura 45: Interpretación del esquema de la zona de laboratorios de EPSL

En la visualización anterior se va estudiar teniendo en cuenta que el Raspberry 1 actúa como nodo central y que este va tener doble función, es decir, funcionara como nodo servidor-cliente y el resto de los nodos actúan como nodos clientes.

4.4.5 Ejemplo de funcionamiento cliente/servidor

En este apartado se muestra la programación que simplifica la interacción entre los nodos clientes con el nodo servidor y viceversa, teniendo en cuenta las especificaciones descritas anteriormente.

Ejemplo de funcionamiento del nodo servidor escrito en nodejs:

```
//Prueba con un servidor que utiliza certificado digital Servidor seguro
```

```
//Necesario para acceder a los certificados en el disco
var fs = require('fs');

//Leemos los archivos de certificado
var privateKey = fs.readFileSync('certificados/key.pem', 'utf8');
var certificate = fs.readFileSync('certificados/cert.pem', 'utf8');
//var ca = fs.readFileSync('certificados/ca-crt.pem', 'utf8');

//Creamos un servidor HTTPS con las credenciales del certificado
var credentials = { key: privateKey, cert: certificate};
var https = require('https');
var httpsServer = https.createServer(credentials);
//Activamos el servidor
httpsServer.listen(8443);

//Asociamos el servidor HTTPS con el servidor de sockets
var WebSocketServer = require('ws').Server;
var wss = new WebSocketServer(
        {
        server: httpsServer
        });


//Procesado de mensajes
wss.on('connection', function connection(ws)
{
        console.log("Conexión con el cliente");
        ws.on('message', function incoming(message)
        {
        console.log(message);
                //Envía un mensaje de vuelta al cliente
        ws.send(message);
        });
        ws.on('close', function close()
        {
                console.log("El cliente ha cerrado la sesión");
```

```
        });
    });
```

Ejemplo de funcionamiento del nodo cliente escrito en nodejs:

```
    //Programa de pruebas para obtener las características de los beacons, utilizando el módulo bleacon
    //y enviando los datos cifrados mediante certificado a un servidor mediante un cliente websocket
        //Módulo para gestión de beacons
        var Bleacon = require('bleacon');
        //Módulo para gestión de websocket
        var WebSocket = require('ws');
        //Definimos el websocket con la dirección y puerto del servidor para conexión no cifrada
        //var ws = new WebSocket('ws://192.168.1.1:8443');
        //Definimos el websocket con la dirección y puerto del servidor para conexión cifrada
        //Usamos el protocolo wss equivalente a https. Como no disponemos de un certificado verificado en el servidor
        //forzamos a que no rechace la conexión; en caso contrario daría un error
        var ws = new WebSocket('wss://192.168.1.1:8443', null, {rejectUnauthorized: false});
        //La identidad del Raspberry en concreto
        var nodeID = "1";
        //Nos indica si estamos o no conectados al servidor
        var conectado = false;
        //La UUID de los beacons con los que vamos a trabajar
        var UUID = '41fffb422be45480be0e29de19634b79';
        var nodeIP = '192.168.1.1';
        var statusMessage = 'stMsg';
        var dataMessage = 'potencia';

        console.log("Esperando al servidor");
        //Este método conecta con el servidor de websocket
        //Una vez arrancada la aplicación, queda esperando hasta que puede conectarse al servidor
        ws.on('open', function open()
        {
            console.log("Conectando al servidor");
```

```
                //Cambiamos a estado conectado
                conectado = true;
                ws.send(statusMessage+";"+nodeID+";"+nodeIP+";"Raspberry);
        });
        //Detectamos si se trata de cerrar la sesión
        ws.on('close', function close()
        {
                console.log("El servidor ha cerrado la sesión");
        });


        //Mensajes recibidos desde el servidor
        ws.on('message', function(message) {
            console.log('Comunicaciones establecidas: %s', message);
                console.log("Esperando detección de beacon");
        });

        //Evento generado cuando hay un error de conexión
        ws.on('connect_error', function (e) {
            console.log('System', e ? e : 'Error desconocido');
         });

        //Arranca el escaneo de beacons
        Bleacon.startScanning();

        //Se activa con cada descubrimiento
        Bleacon.on('discover', function(bleacon) {
         //console.log('bleacon found: ' + JSON.stringify(bleacon));
         //Sólo si estamos conectados al servidor enviaremos datos, naturalmente
         if (conectado)
         {
                //Añadimos comprobación de que el identificador del beacon es el correcto; sólo
en ese caso
                //enviamos datos
                if (bleacon.uuid == UUID)
                {
                        console.log("Beacon correcto, envío de datos");
```

```
                    //Enviamos los datos de interés separados por puntos y comas

ws.send(dataMessage+";"+bleacon.major+";"+bleacon.minor+";"+bleacon.rssi+";"Raspberry);

console.log(dataMessage+";"+bleacon.major+";"+bleacon.minor+";"+bleacon.rssi+";"Raspberry);
                }
            }
            //console.log(bleacon.rssi);
        });
```

4.4.6 *Análisis de datos y resultados*

Cada análisis de datos tiene algunos componentes estándar:

- Preprocesamiento: consideración de los datos obtenidos de la aplicación para identificar posibles valores procedentes en cada nodo cliente.
- Resumen: cálculo de estadísticas para describir la ubicación y forma generales de los datos.
- Visualización: graficación de datos para identificar el nodo con mayor potencia recibida.

En este apartado se explica, cómo llevar a cabo un análisis básico en entorno MATLAB.

Preprocesamiento de datos:

En este apartado se muestra como preprocesar datos con fines de análisis:

- **Descripción general**: comencemos analizar los datos cargando los datos en variables adecuadas de MATLAB y considerando los datos de interés entre los datos recibidos. Este es un paso preliminar que ayuda asegurar que se llegue a conclusiones significativas en partes posteriores de análisis.
- **Carga de los datos**: Cargamos los datos con la función **fopen** de matlab.
- **Datos faltantes**: El valor NaN (No es Número) de MATLAB normalmente se usa para representar datos faltantes. Los valores NaN permiten que las variables con datos faltantes mantengan su estructura (en nuestro caso cuando el iBeacon está lejos de algunos de los nodos clientes, apenas recibimos los datos de este porque no se comunican entre sí, en este caso los valores saldrán de este tipo).

- **Calculo de la potencia media**: Calculamos la potencia media por cada número de muestras de los datos recibidos en cada Raspberry (Nodos clientes) para identificar el nodo con mayor o menor intensidad recibida, la función **mean** de matlab nos permite hacer esta operación.
- **Visualización**: Se muestra gráficamente los niveles de potencias medias aplicadas en cada ejecución, lo que nos permitirá a identificar con más facilidad el nodo con mayor intensidad recibida.

Código fuente:

```
%------------------------procesado de información-----------------------
---

%este método consiste en separar tres Raspberry en tres laboratorios para
%medir la intensidad de la señal recibida, con lo cual nos permite
%determinar el Raspberry que mejor recibe en cada ubicación de
%iBeacon, siendo este último dispositivo el sensor que envía los datos de
%de forma cifrada a todos los nodos clientes, en nuestro caso hemos
cogido las muestras en los siguientes laboratorios:
% --laboratorio de radiofrecuencia y microondas->Rb1->actuando como
cliente y servidor
% --laboratorio de servicios y sistemas->Rb2->actuando como cliente
% --laboratorio de sistemas telemáticos->Rb3->actuando como cliente
%Y por último los datos se han guardado en el siguiente orden:
% --laboratorio de radiofrecuencia y microondas->servidor recibe los
datos procedentes de Rb1,Rb2,Rb3, siendo el nombre del archivo
entonces:datosRF.txt
% --laboratorio de servicios y sistemas->servidor recibe los datos
%procedentes de Rb1, Rb2,Rb3, siendo el nombre de archivo
entonces:datosSS.txt
% --laboratorio de radiofrecuencia y microondas->servidor recibe los
datos
%procedentes de Rb1,Rb2,Rb3, siendo el nombre del archivo
entonces:datosST.txt

clear all;
% Cargar datos
file = fopen('datosRF.txt');%cambiando el nombre del archivo de entrada
c = textscan(file,'dat;1;2;%d;%s');
fclose(file);

% Adaptar datos:
pot = c{1,1};  %potencia
nom = c{1,2};  %nombres:Rb1,Rb2,Rb3
Id = cell2mat(nom); %Identificador
Id = str2num(Id(:,3));

% Posiciones
pos_Rb1 = find(Id == 1);
pos_Rb2 = find(Id == 2);
pos_Rb3 = find(Id == 3);
% Valores de potencia
pot_Rb1 = pot(pos_Rb1);
```

```
pot_Rb2 = pot(pos_Rb2);
pot_Rb3 = pot(pos_Rb3);
% Potencia media de datos
Pm_Rb1 = mean(pot_Rb1);
Pm_Rb2 = mean(pot_Rb2);
Pm_Rb3 = mean(pot_Rb3);
%guardamos las potencias medias en forma vectorial
%pot_med = [Pm_Rb1; Pm_Rb2; Pm_Rb3];
%Representamos las potencias medias
%stem(pot_med)
```

4.4.7 *Resultados de la aplicación*

En este apartado se presentan los resultados obtenidos de la aplicación de descrita anteriormente, concretamente los niveles de potencia que procesa el servidor en cada ejecución.

El método consiste en calcular las potencias medias para verificar los nodos con mayor o menor intensidad recibida en escalas logarítmicas (dBm ó RSSI) en cada posición que se encuentra el iBeacons. El código matlab descrita anteriormente grafica histogramas de las potencias medias de las muestras recogidas para visualizar las siguientes figuras:

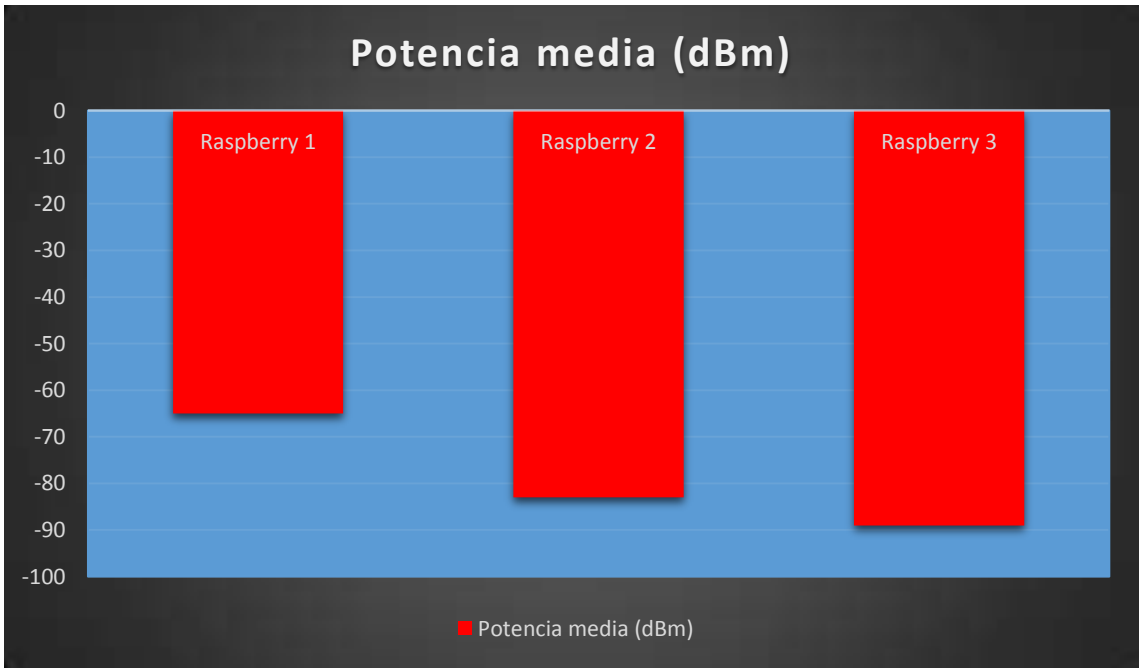


Figura 46: Valores de las potencias medias de la simulación de laboratorio de RF

En la visualización anterior, se observa claramente que el Raspberry 1 contiene mayor intensidad recibida con un valor aproximadamente de -65dBm,mucho mayor que las potencias medias del resto de los nodos clientes, como ha de esperar porque en esta

simulación el iBeacons se encuentra situado dentro del laboratorio de Radiofrecuencia y Microondas (RF) que coincide con la posición del Raspberry 1 , generalmente la escala se expresa dentro de valores negativos; cuanto menos negativo, menos pérdidas de señal.

A continuación pasamos analizar las potencias medias con otros tipos de valores de la simulación para visualizar la siguiente figura:

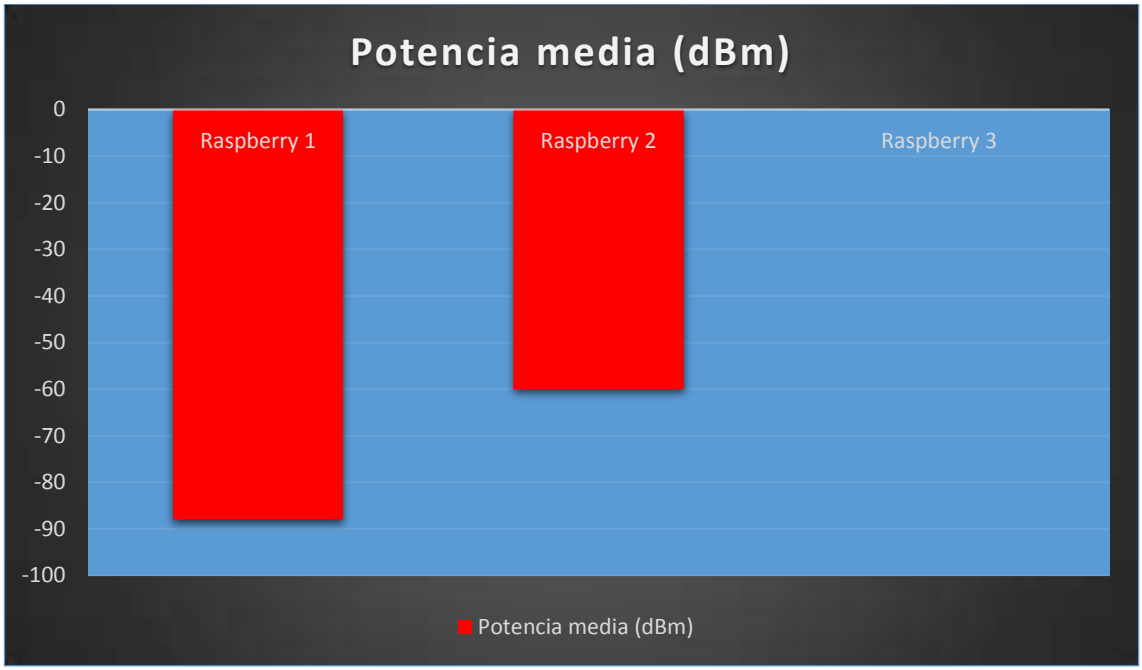


Figura 47: Valores de las potencias medias de la simulación de laboratorio de SS

En la figura 47, se observa que el Raspberry con mayor intensidad recibida es el 2, que coincide con la ubicación de iBeacons y el Raspberry con menos intensidad recibida es el Raspberry 1, apenas recibe nada el Raspberry 3 porque se encuentra muy alejado de la posición de iBeacons, es decir esta fuera de área de cobertura y no puede comunicar con el Raspberry 2, en este caso solo podría comunicar con el Raspberry 2 pasando por el Raspberry 1.

Finalmente pasamos analizar los valores para el caso cuando el iBeacons se encuentra situado en el laboratorio de Sistemas Telemáticos, que coincide con la ubicación de Raspberry 3, con lo cual visualizamos la siguiente imagen:

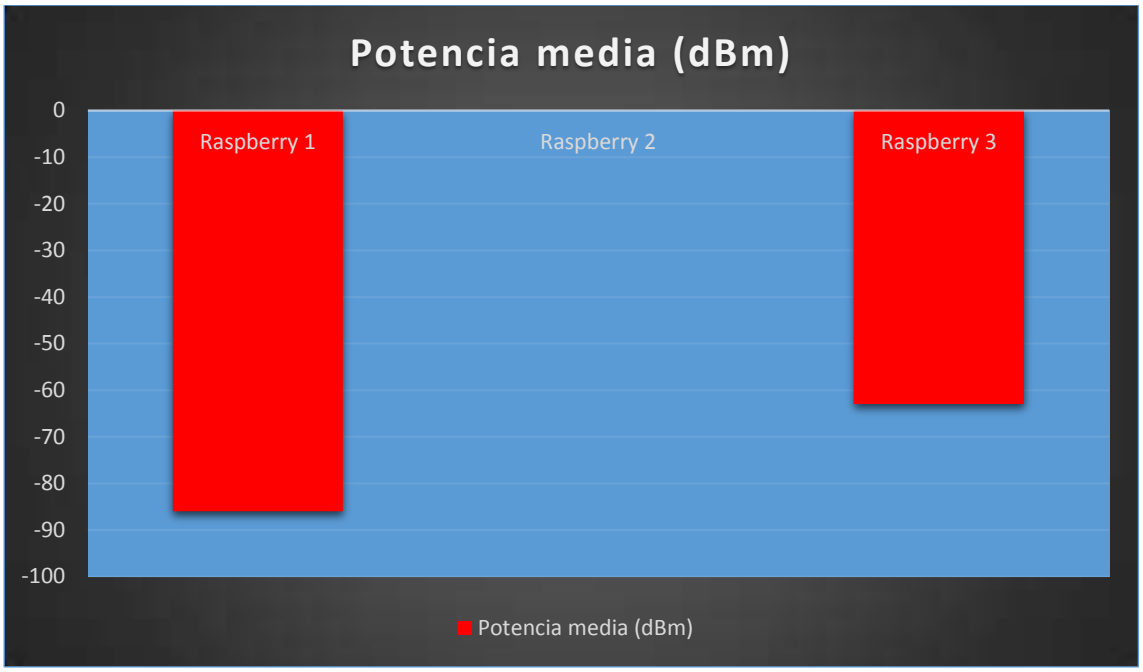


Figura 48: Valores de las potencias medias de la simulación de laboratorio de ST

Al igual que en el caso anterior, con la diferencia de que ahora el Raspberry con mayor intensidad recibida es el 3 y el que menos intensidad recibe es el Raspberry 1, siendo el peor dispositivo en este caso, el Raspberry 2.

5 CONCLUSIONES

El objetivo final del trabajo era desarrollar un sistema de microlocalización y seguimiento en interiores con sensores Bluetooth LE sobre una red MANET, para procesar niveles de potencia de las señales recibidas por los iBeacons en una red MANET y, a su vez, transmitir adecuadamente y de forma cifrada esta información a un sistema central de procesado de información. Esa aplicación ya es una realidad.

En primer lugar destacar que, las MANETs son redes ilimitadas, es decir, que no solamente son operables para aplicar técnicas de localización en interiores, también pueden ser utilizadas en forma autónoma o como una opción complementaria a las redes convencionales y estar conectada a una red celular o internet, por lo tanto, se pueden utilizar para operaciones de rescate y en entornos militares, por lo cual, las MANETs nos facilitan múltiples retos que deben ser resueltos para que puedan ser completamente aprovechadas en aplicaciones comerciales.

En segundo lugar destacar que los nodos de la red MANETs, han sido los dispositivos Raspberry Pi 3B que ofrece mejores prestaciones, con menos coste y robustas en el uso de redes MANETs para la prestación de servicios como Bluetooth y acceso a internet público.

En tercer lugar destacar que, las MANETs funcionan con protocolos de enrutamiento, en este proyecto se ha empleado el protocolo OLSRD para aprovechar que las rutas de todos los destinatarios de la red sean conocidas y mantenidas antes de su uso sin sufrir el retardo de descubrimiento de ruta asociada a la búsqueda de una nueva ruta y, además por cada nodo que se inserta, permite una nueva actualización en la tabla de encaminamiento de forma automática.

En cuarto lugar comprobamos que, los protocolos de seguridad funcionan correctamente salvando las vulnerabilidades por lo cual obtenemos los resultados significativas.

En quinto lugar destacar que los beacons son dispositivos muy destacados en la actualidad desde el punto de vista de las TIC y a nivel de negocio, porque involucra para dar soluciones en otros ámbitos como envíos in site, microsegmentación, seguimiento de usuarios por su ubicación o proximidad al local, Big data, navegación indoor, atención al cliente en tienda, etc.

6 LINEAS FUTURAS

En lo que concierne a las líneas de investigación futura, durante el proceso de elaboración de este proyecto se ha considerado interesantes los temas que se exponen a continuación.

En primer lugar, resultaría más cómodo por cualquier persona o un administrador de red conocer los mapas de potencia e intensidades de las señales WiFi, ya sea de una casa, de una empresa, lugar de trabajo, etc, por lo tanto, sería interesante incrementar la cantidad de beacons y modificar el programa que permitirá a visualizar un mapa que muestra las áreas y zonas con más potencia y las zonas con señal más débil.

En segundo lugar, resultaría muy útil en grandes empresas que todos los departamentos tengan acceso a internet con buena calidad de la señal sin perdidas, sería interesante incrementar la cantidad de Raspberry Pi 3B para ampliar la señal WiFi y ahorrar el coste a nivel de la infraestructura de la red inalámbrica.

7 REFERENCIAS BIBLIOGRAFICAS

[1] Conceptos y características de las redes WiFi, disponible en:

Cisco networking academy program. Edición: 3ª ed. Autor:.-Editorial:
Madrid: Pearson Educación, D.L. 2004
http://bibing.us.es/proyectos/abreproy/11761/fichero/Volumen1%252F5-Cap%C3%ADtulo1+-+Introducci%C3%B3n+a+las+redes+inal%C3%A1mbricas.pdf+
https://es.wikipedia.org/wiki/Wifi
https://en.wikipedia.org/wiki/List_of_WLAN_channels#/media/File:2.4_GHz_Wi-Fi_channels_(802.11b,g_WLAN).svg
https://en.wikipedia.org/wiki/IEEE_802.11a
http://www.eveliux.com/mx/ancho-de-banda-definicion.html
https://gl.wikipedia.org/wiki/Ancho_de_banda
https://en.wikipedia.org/wiki/List_of_WLAN_channels

[2] Conceptos y características de redes Mesh, disponible en:

http://bibing.us.es/proyectos/abreproy/11306/fichero/TEORIA%252F08+-+Capitulo+3.pdf
https://es.wikipedia.org/wiki/Red_ad_hoc_inal%C3%A1mbrica
http://www.redalyc.org/pdf/4026/402640449002.pdf
https://es.scribd.com/document/53148802/Redes-Ad-Hoc

[3] configuración de la red ad-hoc, disponible en:

https://geekytheory.com/tutorial-rasbperry-pi-como-crear-un-punto-de-acceso-wifi
https://www.redeszone.net/raspberry-pi/manual-para-configurar-raspberry-pi-como-un-router-wi-fi/
https://www.alvarolara.com/2014/08/29/punto-de-acceso-wifi-en-raspberry-pi/

[4] Conceptos y configuración del protocolo OLSRD

https://es.wikipedia.org/wiki/Optimized_Link_State_Routing
https://www.raspberrypi.org/forums/viewtopic.php?t=103550
http://www.tldp.org/HOWTO/OLSR-IPv6-HOWTO/

[5] Configuración de servicios Bluetooth, disponible en:
https://www.raspberrypi.org/forums/viewtopic.php?f=36&t=26826
https://nodejs.org/en/download/package-manager/#debian-and-ubuntubased-linux-distributions
https://blog.miniarray.com/installing-node-js-on-a-raspberry-pi-zero-21a1522db2bb

[6] Instalación y programación en nodejs, disponible en:

https://nodejs.org/en/download/package-manager/#debian-and-ubuntubased-linux-distributions
http://zempirians.com/ebooks/Basarat%20Ali%20Syed%20(auth.)-Beginning%20Node.js-Apress%20(2014).pdf

[7] protocolos de la capa de aplicación, disponible en:

http://www.csd.gob.es/csd/instalaciones/descripcion-de-la-aplicacion-informatica
http://www.ajmuro.net/documents/generar-pem.pdf
http://www.conta5.com/FAQS/Generar_fichero_PEM_Certificado.pdf
https://es.wikipedia.org/wiki/Base64
https://es.wikipedia.org/wiki/WS-Securit
http://cleventy.com/paginas-web-seguras-que-es-el-https/
https://stackoverflow.com/questions/29047078/how-to-enable-https-localhost-url-in-wamp-server-v2
https://es.wikipedia.org/wiki/WebSocket
https://victordiaz.me/websocket

[8] Información sobre iBeacons, disponible en:

http://www.bihartech.com/ibeacon/
https://es.wikipedia.org/wiki/IBeacon
https://www.youtube.com/watch?v=O_Ka7jdxFH0
https://accent-systems.com/es/producto/ibks-105/?gclid=EAIaIQobChMIkr6Cj9DK2AIVwgrTCh2f_g6kEAAYASAAEgIQovD_BwE
https://accent-systems.com/wp-content/uploads/iBKS105_datasheet_rev3.pdf
https://www.easycontext.com/ventajas-uso-beacons-sector-retail

[9] Instalación de Raspbian y WinsCP, disponible en:
http://dau-sa.dst.usb.ve/manuales/winscp.pdf
https://camins.upc.edu/ca/pdf/pdf-serveis/recursos-per-a-la-recerca/manual-de-winscp
https://www.raspberrypi.org/downloads/

[10] programación en Matlab, disponible en:

https://es.mathworks.com/help/matlab/learn_matlab/data-analysis.html
https://es.mathworks.com/help/matlab/ref/fopen.html

8 ANEXOS

8.1 ANEXO 1

Instalar sistema operativo (Raspbian NOOBS).

Recursos necesarios:

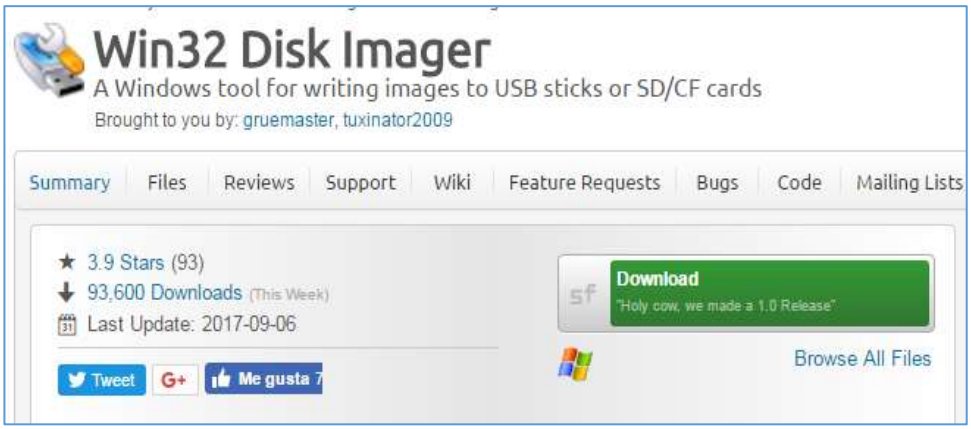
- Tarjeta micro SD de 8GB
- Una computadora con lector de tarjeta micro SD y sistema operativo Windows 7.

Software necesario:

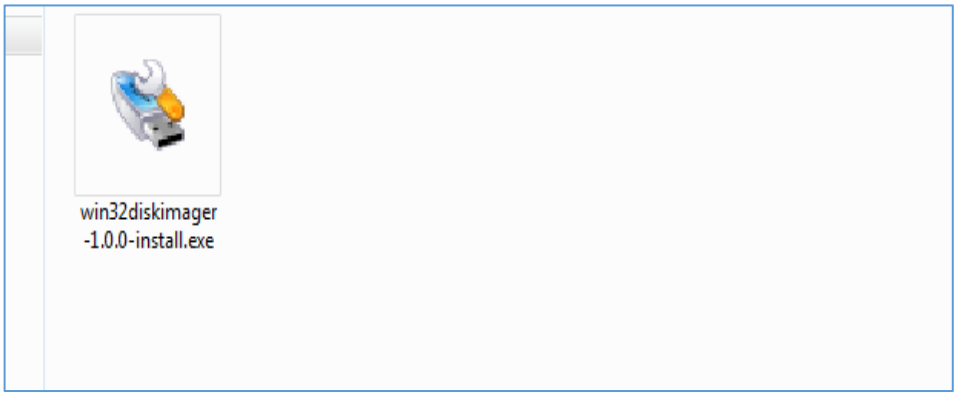
- Archivo con la imagen de Raspbian NOOBS
- Win32 Disk Imager versión 0.9.5

Paso 1: Descargar y descomprimir

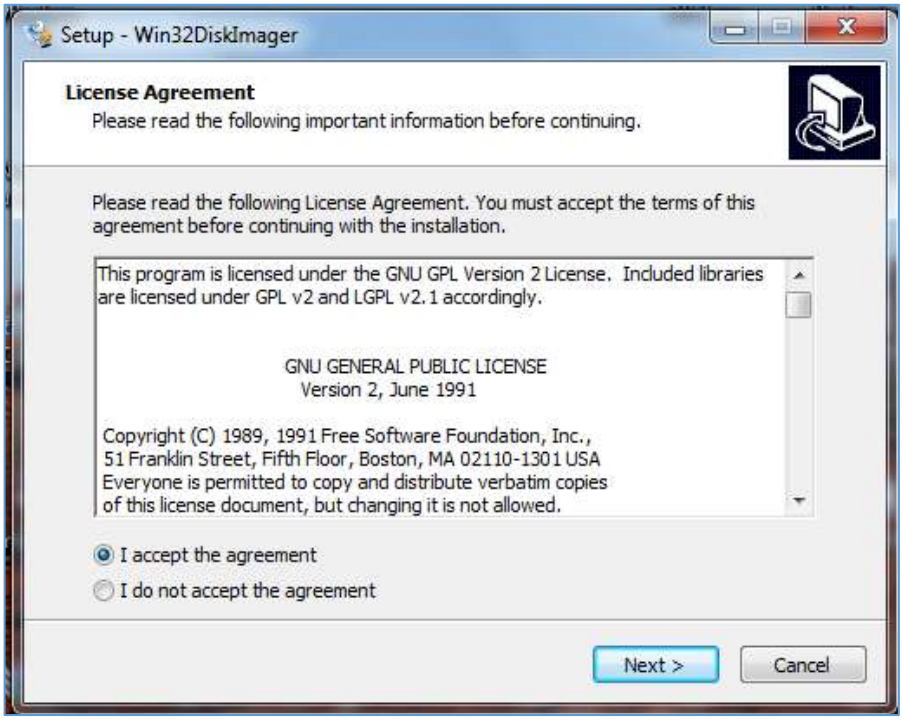
Ingresamos la siguiente URL: https://www.raspberrypi.org/downloads/noobs/ y descargamos el archivo de la imagen de Raspbian en un ordenador, una vez descargando descomprimido el archivo, lo guardamos en una ubicación conocida para su uso posterior.



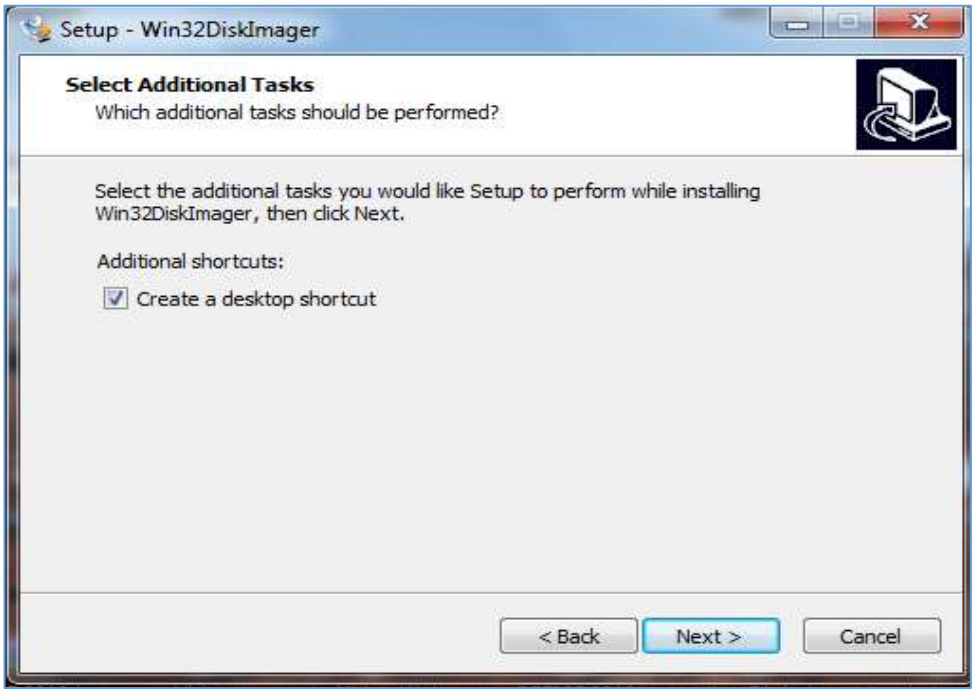
Abrimos el archivo descargado, de preferencia y lo ejecutamos como administrador



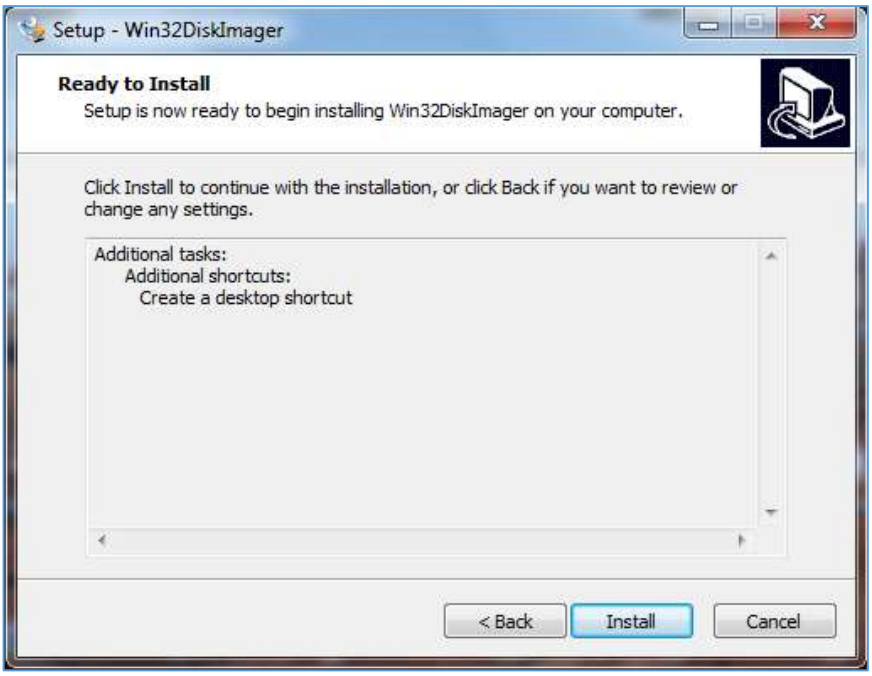
Presionamos aceptar a los términos de instalación del Software y continuar.



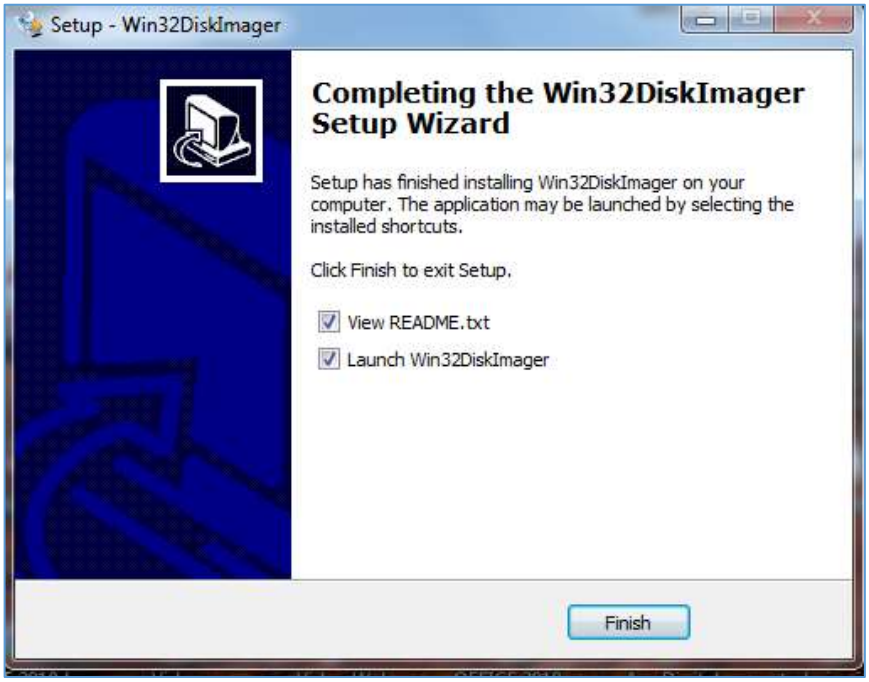
Escogemos el directorio en el que va a instalar el programa, se puede dejar con el directorio por defecto.



Luego de estos pasos se mostrará una pantalla con la confirmación de las preferencias escogidas, presionamos instalar para proceder con la instalación.



Se mostrará una pantalla con la confirmación de la instalación en la que se podrá escoger si se desea leer el archivo de la post instalación o si lo deseamos ejecutar directamente, escogemos lo que desea hacer y presionamos finalizar.



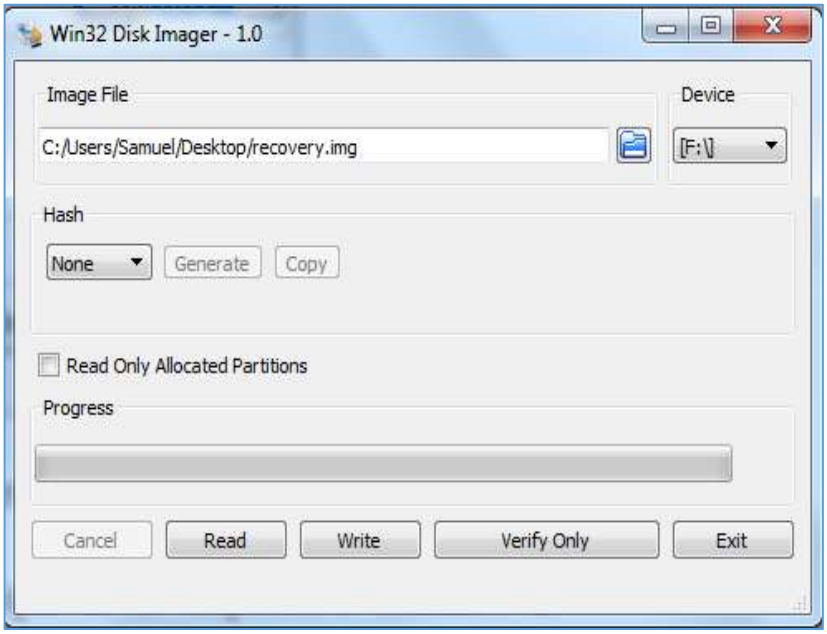
Insertamos la memoria SD en el equipo, se puede hacer con el lector SD o bien adaptador USB para que se identifique la unidad que se le asigno. Para conocer la unidad entramos en equipo e identificar la unidad que le corresponde a la memoria SD tal como se muestra en la imagen. En “Equipo” (Clic en inicio y luego “Equipo”) se puede observar todos los medios de almacenamiento interno y externos.

- Ejecutar Win32 Disk Manager.
- Clic en la imagen de la carpeta.

En el recuadro emergente buscamos el archivo de la imagen de Raspbian que descargo y luego clic en “abrir”.

Debajo de la opción “Device” despliegue la lista de dispositivos disponibles y seleccionamos la letra que identifica a la memoria SD previamente reconocido.

Una vez seleccionado el dispositivo correcto clic en “Write”.



En la advertencia que aparecerá damos clic en “Yes”.

Finalmente grabamos la imagen en la memoria micro SD y asegurando que la grabación ha culminado con éxito. La memoria micro SD ahora contiene el sistema operativo necesario para la Raspberry Pi 3B.

8.2 ANEXO 2

Instalar Winscp.

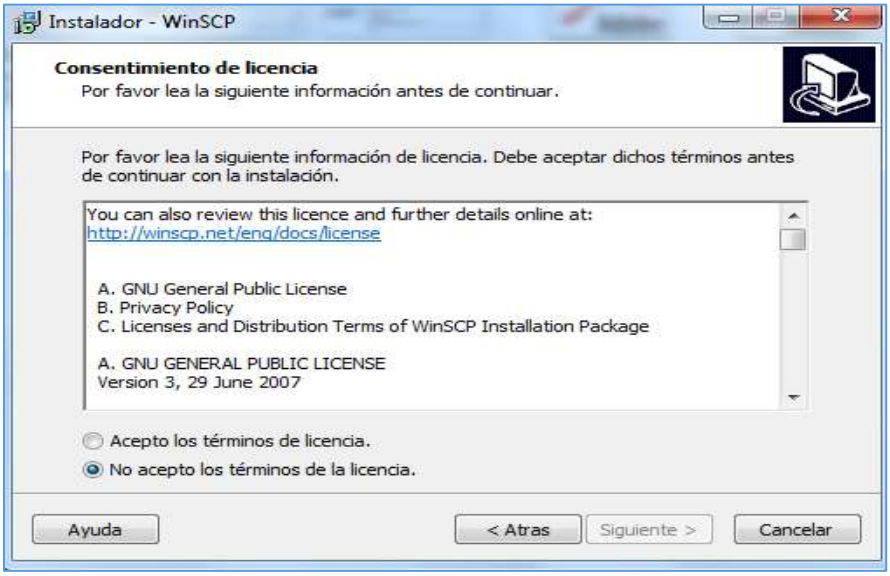
WinSCP es un programa que nos permite transferir ficheros, de forma segura, de nuestro ordenador hasta una máquina remota.

Instalación.

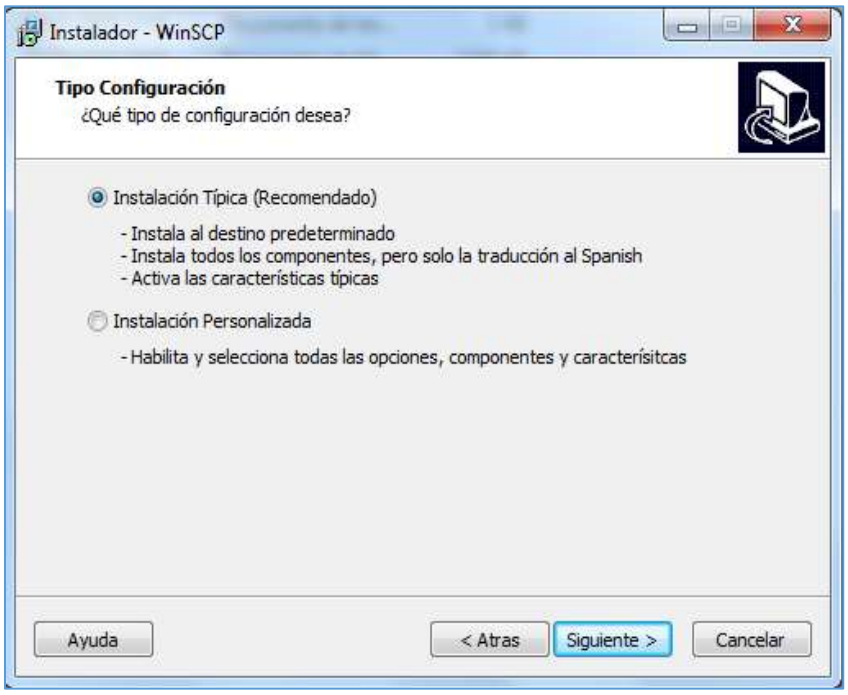
Ingresamos la siguiente URL: https://winscp.softonic.com/

Una vez descargado el archivo, abrimos el asistente de instalación y lo ejecutamos como administrador.

Pulsamos el botón siguiente para continuar

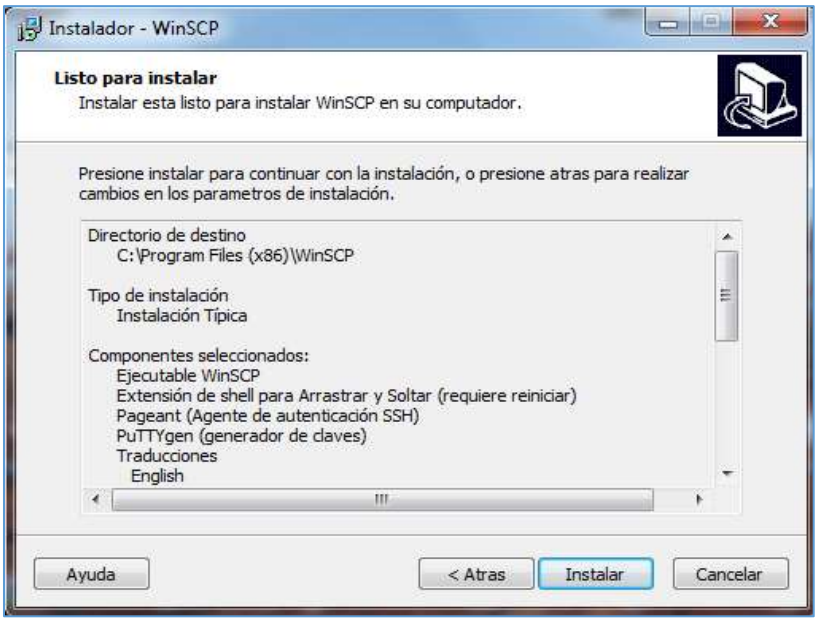


Aceptamos los términos de licencia y le damos otra vez el botón siguiente para continuar.



Pulsamos el botón siguiente para continuar.

Pulsamos el botón siguiente para continuar.

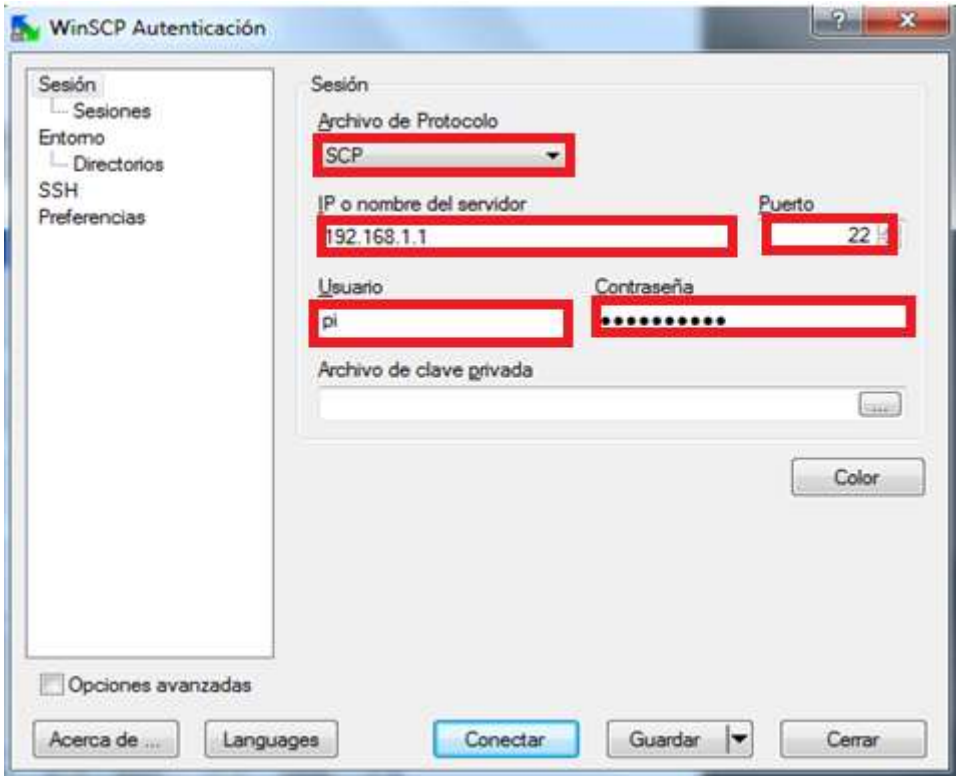


Pulsamos el botón instalar para iniciar la instalación.

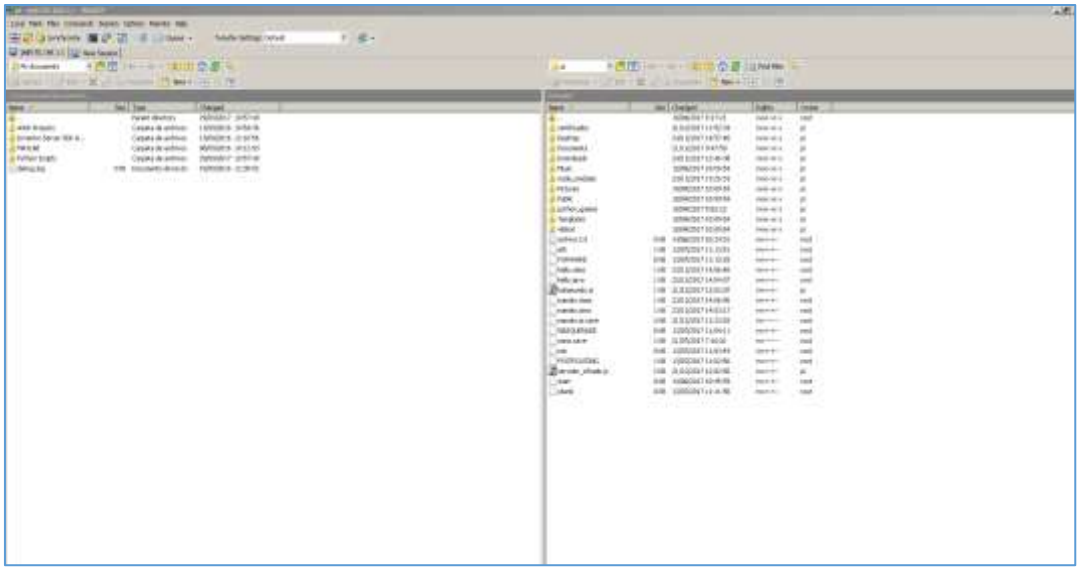
La instalación ha finalizado. Se puede ejecutar desde el menú Inicio Programas o desde el icono de acceso directo en el escritorio.

Conexión.

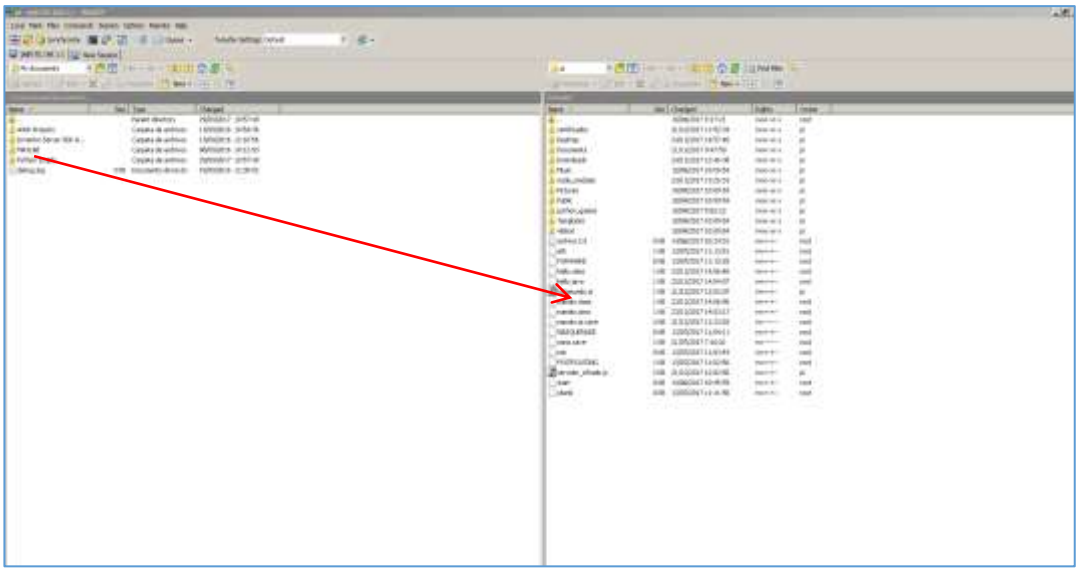
Para iniciar una conexión, primero ponemos el nombre de la máquina (Host name), Usuario y contraseña correspondientes. A continuación, pulsamos guardar.



Una vez se ha conectado, se muestra una pantalla como la siguiente. En la parte izquierda podemos navegar por los archivos del equipo y a la derecha tenemos los de la máquina remota.



Finamente para transferir archivos de un lugar a otro basta con arrastrarlos de un área a la otra.



8.3 ÍNDICE DE FIGURAS

8.4 ACRÓNIMOS

GPS: Global Positioning System (Sistema de Posicionamiento Global)

MANET: Mobile Ad-hoc NETwork

OLSRD: Optimized Link State Routing Deamon (Demonio de Enrutamiento de Estado de Enlace)

WLAN: Wireless local area network (red de área local inalámbrica)

IEEE: Institute of Electrical and Electronics Engineers (Instituto de Ingenieros Eléctricos y Electrónicos)

PSK: Phase Shift Keying (modulación por desplazamiento de fase)

QPSK: Quadrature phase-shift keying (modulación de desplazamiento de fase en cuadratura)

OFDM: Orthogonal Frequency Division Multiplexing (Multiplexación por División de Frecuencias Ortogonales)

QAM: Quadrature Amplitude Modulation (modulación de amplitud en cuadratura)

WAP o AP: Wireless access point (punto de acceso inalámbrico)

SSID: Service Set Identifier

DHCP: Dynamic Host Configuration Protocol (protocolo de configuración dinámica de host)

MAC address: media access control address

WEP: Wired Equivalent Privacy (Privacidad equivalente a cableado)

WPA: Wi-Fi Protected Access (Acceso Protegido Wi-Fi)

WPA2: Wi-Fi Protected Access 2 (Acceso Protegido Wi-Fi 2)

PCI: Peripheral Component Interface (interface para componentes periféricos)

USB: flash drive e pen drive (Universal Serial Bus)

GPRS: General Packet Radio Service (servicio general de paquetes vía radio)

UMTS: Universal Mobile Telecommunications System (Sistema Universal de Telecomunicaciones Móbiles)

HDMI: High-Definition Multimedia Interface (interfaz multimedia de alta definición)

CSI: Camera Serial Interface (Interfaz Serie para Cámaras)

DSI: Display Serial Interface (Interfaz serie de pantalla)

IPS: Indoor Positioning System (sistema de posicionamiento interior)

ESSID: Extended Service Set Identifier

URL: Uniform Resource Locator (localizador uniforme de recursos)

SSH: Secure SHell (intérprete de órdenes seguro)

LAN: Local Area Network

IP: Internet Procol

MPR: Retransmisores multipunto

MacOS: Macintosh Operating System (Sistema Operativo de Macintosh)

WSS: WS-Security (Seguridad en Servicios Web)

HTTP:Hypertext Transfer Protocol

HTTPS:Hypertext Transfer Protocol Secure

SSL: Secure Sockets Layer

ASCII: American Standard Code for Information Interchange (Código Estándar Estadounidense para el Intercambio de Información)

ISO: International Organization for Standardization (Organización Internacional de Normalización)

IEC: Institut d'Estudis Catalans (Instituto de Estudios Catalanes)

TCP: Transmission Control Protocol (Protocolo de control de transmisión)

RSSI: Received Signal Strength Indicator (indicador de fuerza de la señal recibida)

WinSCP: Windows Secure Copy

TIC: Tecnologías de Información y Comunicación