Universidad Carlos III de Madrid

Escuela Politecnica Superior

Ingeniería de Telecomunicación

Proyecto Fin de Carrera

Diseño, despliegue y evaluación experimental de una red inalámbrica multi-salto IEEE 802.11g

Autor: Miguel Angel Flores Trueba

Tutor: Carlos J. Bernardos Cano

Julio 2010

PROYECTO FIN DE CARRERA

Departamento de Ingeniería Telemática

Universidad Carlos III de Madrid

Titulo: Despliegue y experimentación en una red mesh 802.11

Autor: Miguel Angel Flores Trueba

Tutor: Carlos J. Bernardos Cano

La lectura y defensa de presente proyecto fin de carrera se realiza el 28 de Julio de 2010 bajo el tribunal:

- **Presidente:**
- **Secretario:**
- **Vocal:**

Habiendo obtenido la calificacion de:

Presidente **Secretario** **Vocal**

Agradecimientos

Índice general

Resumen **11**

I Introducción **14**

1 Introducción **15**

1.1 Motivación del Proyecto 15

1.2 Objetivos del Proyecto 16

1.3 Estructura de la memoria 16

II Estado del Arte **18**

2 Redes Inalámbricas **19**

2.1 Comunicaciones Inalámbricas 19

2.2 Modos de operación de redes inalámbricas 19

2.2.1 Red Ad-Hoc 20

2.2.2 Infraestructura 20

2.2.3 Red inalámbrica Mesh 22

2.3 Despliegues de redes Mesh 25

2.3.1 Redes mesh comunitarias 26

2.3.2 Redes mesh comerciales 27

2.3.3 Redes mesh de laboratorio 28

2.4 Estándares IEEE 802.11 30
2.4.1 IEEE 802.11a 30
2.4.2 IEEE 802.11b 31
2.4.3 IEEE 802.11g 31
2.4.4 IEEE 802.11n 32
2.4.5 IEEE 802.11s: Mesh 32
2.5 Conclusiones 33

III Trabajo realizado 34

3 Despliegue de la Red de Pruebas 35
3.1 Diseño de la red 35
3.1.1 Requisitos 35
3.1.2 Descripción del Entorno 36
3.1.3 Descripción de la red 37
3.1.4 Coste de los equipos 42
3.2 Instalación y Configuración 43
3.2.1 Herramientas utilizadas 45
3.3 Conclusiones 47

4 Evaluación Experimental 48
4.1 Diferencias entre generar trafico en PC y routers 49
4.1.1 Asus 802.11g vs. PC 50
4.1.2 Fonera 802.11g vs. PC 51
4.2 Estudio del efecto aislante del subsuelo 53
4.3 Impacto de la hora del día 56
4.4 Impacto de la potencia de transmisión 60
4.5 Impacto de la interferencia en canales 802.11b/g adyacentes . 61

4.5.1 Escenario A: interferentes lejanos 63
4.5.2 Escenario B: interferentes cercanos 68
4.6 Medidas en red Multisalto 73

5 Conclusiones y trabajos futuros 82
5.1 Conclusiones 82
5.1.1 Red de pruebas 82
5.1.2 Conclusiones experimentales 83
5.2 Trabajos futuros 85

IV Apéndices 86

A Presupuestos y diagrama de tareas 87
A.1 Introducción 87
A.2 Presupuesto del Proyecto 88
A.3 Diagrama de Gantt 89

B Anexos 91
B.1 Anexo: Cómo Instalar OpenWrt en un router ASUS WL-500g Premium 91
B.2 Anexo: Cómo instalar OpenWrt en una Fonera (modelo 2100) 97
B.2.1 Incompatibilidades de firmwares en las foneras: 107
B.2.2 Flasheando la fonera: 108
B.2.3 Dejar una fonera 2100 con los valores de fábrica . . . 108
B.3 Anexo: Montar servidor tftpd 111
B.4 Anexo: Herramientas 112
B.4.1 Manual iperf 112
B.4.2 Manual nagios 114
B.4.3 Manual tcpdump 118

B.4.4 Otras herramientas y comandos 125
B.5 Anexo: Diagramas de flujo 127
B.5.1 Script generar tráfico en PC o en routers 127
B.5.2 Script del impacto de la hora del día 128
B.5.3 Script de filtrado de canales wifi 129
B.6 Ayudas y manuales referenciados en los anexos 130
B.7 Anexo: Tabla de direccionamiento y ubicación de equipos . . 131
B.8 Anexo: Plano del laboratorio 132

Referencias **133**

Índice de figuras

2.1 Red Ad Hoc (Fuente: http://www.debahia.com) 20

2.2 Red Infraestructura(Fuente: http://www.debahia.com) 21

2.3 Ejemplo red Mesh 24

2.4 Red inalámbrica mesh con infraestructura 25

2.5 Red inalámbrica mesh con clientes 25

2.6 Red inalámbrica mesh híbrida 26

2.7 Aplicación práctica de red mesh en área metropolitana 28

2.8 Topología de la red CARMEN 30

2.9 Canales en 802.11a 31

2.10 Canales en 802.11b/g 32

3.1 Situación de los nodos en la red de pruebas del laboratorio . . 38

3.2 Linksys WRT54GL 39

3.3 Asus WL-500g 40

3.4 Fonera 40

3.5 Fotografía de un nodo de la red de pruebas 41

3.6 Situación de elementos bajo las losetas 42

3.7 Esquema de red, topología lógica cableada e inalámbrica . . . 45

4.1 Impacto de generar tráfico desde PC o desde los dispositivos Asus en modo 802.11g 50

4.2 Impacto de generar tráfico desde PC o desde las foneras, 802.11g 51

4.3 Detalle de loseta del falso suelo 53
4.4 Ambos routers bajo el suelo 54
4.5 Un router encima del suelo, y otro debajo 54
4.6 Ambos routers encima del suelo 55
4.7 Efecto del aislamiento eléctrico del subsuelo 55
4.8 Impacto de la hora del día en routers Asus 802.11g 57
4.9 Rendimiento en función de los canales en Asus 802.11g 59
4.10 Impacto de la potencia de transmisión en la red de pruebas con Asus 802.11g 61
4.11 Escenarios de la prueba 62
4.12 Enlaces radiando por separado con Asus 802.11g en el escenario A 64
4.13 Enlaces radiando juntos con Asus 802.11g en el escenario A . 65
4.14 Separación entre canales d=0,3,5,10 66
4.15 Interferentes lejanos 802.11g: (η) eficiencia de la separación de canales 67
4.16 Enlaces radiando por separado con Asus 802.11g en el escenario B 69
4.17 Enlaces radiando juntos con Asus 802.11g en el escenario B . 70
4.18 Interferentes cercanos 802.11g: (η) eficiencia de la separación de canales 72
4.19 Escenario de red multisalto con 3 saltos. Asus 802.11g 73
4.20 Escenario de red multisalto con 7 saltos inalámbricos. Asus 802.11g 74
4.21 Medida del ancho de banda en red multisalto con 3 saltos . . . 75
4.22 Síntesis del Algoritmo Heurístico utilizado en esta prueba . . . 77
4.23 Rendimiento en función del número de saltos de la red 80

B.1 Puente entre puertos WLAN y LAN 94

B.2 Esquema conversor puerto serie para fonera 109
B.3 Conexiones internas de la fonera 110
B.4 Diagrama de Flujo de la prueba 4.1 127
B.5 Diagrama de Flujo de la prueba 4.3 128
B.6 Diagrama de Flujo de la prueba 4.3 129
B.7 Plano del Laboratorio 4.1.F04 132

Índice de tablas

3.1 Desglose de los Costes del equipamiento principal del proyecto. 43

4.1 Estadísticos de la prueba de 24 horas 59

4.2 Resultados de las repeticiones del algoritmo heurístico incrementando el número de enlaces 79

B.1 Disposición y direccionamiento de los equipos del proyecto . . 131

Resumen

En la actualidad se están realizando un gran número de proyectos de investigación relacionados con las redes multisalto inalámbricas. La mayoría de estos trabajos se han centrado tradicionalmente en estudios teóricos y/o de simulación, por lo que es necesario avanzar en la línea de plataformas experimentales que permitan validar los resultados teóricos y probar nuevos diseños y algoritmos.

El enfoque que se le da a estos proyectos, en la mayoría de los casos, motiva la creación de plataformas de pruebas, desarrolladas tanto en interiores como en exteriores. Estas redes de pruebas, que requieren una significativa cantidad de recursos a la hora del despliegue y el mantenimiento, se utilizan para llevar a cabo medidas con el fin de analizar y entender las limitaciones y diferencias entre los resultados de los análisis o simulaciones y los obtenidos en experimentaciones reales.

En este proyecto se realizan dos importantes aportes bastante novedosos; en primer lugar se describe un innovador banco de pruebas multisalto inalámbrico, el cual se desarrolla y se opera bajo el suelo de un laboratorio de la universidad. Este falso suelo proporciona una protección frente a otras señales inalámbricas, gracias a las características que poseen los paneles del suelo (durante el desarrollo del proyecto se estudiará más a fondo esta particularidad).

Por otra parte, realizando experimentos controlados, hemos sido capaces de analizar al límite el comportamiento de dispositivos comerciales, así como observar criterios de diseños prácticos para el desarrollo de redes inalámbricas mesh.

Los resultados que hemos obtenido en las experimentaciones de este proyecto, caracterizan los dispositivos inalámbricos utilizados, sientan las bases para futuros experimentos que los empleen y nos permiten obtener conclusiones importantes a la hora de realizar futuros experimentos en esta plataforma u otras similares.

Abstract

Currently, a lot of researches related with multihop wireless networks are being carried out. Most of these projects are focused on theoretical researches and/or simulations, so new experimental testbeds that allow to validate theoretical results and try new algorithms are needed.

The approach to these projects, in most cases, motivates the testbeds, developed both inside and outside. These testbeds require a lot of resources in the deployment and maintenance, and are used to carry out measurements to analyze the limitations and differences between the results from analysis or simulations and the ones obtained in real experimentation.

In this project two innovate important contributions are carried out; first an innovate wireless multhop testbed is described. This testbed is developed and operated under the false floor of lab of a university building. This false floor provides a strong physical against another wireless signals thanks to the false floor panels characteristics (this aspect will be more developed later).

Second, by running controlled experiments we are able to analyze the performance limits of commercial devices, as well as to derive practical design criteria for the deployment and configuration of mesh networks.

The results obtained in the experimentations characterize the wireless devices used, lay the foundations for future tests with these devices and allow us to extract important conclusions from the experiments realized.

Parte I

Introducción

Capítulo 1

Introducción

1.1 Motivación del Proyecto

Actualmente existen algunos proyectos basados en tecnologías inalámbricas 802.11. Algunos de estos proyectos se han centrado en desarrollar y experimentar con redes inalámbricas interiores [2], [3], otros han hecho lo propio con redes desplegadas en el exterior [4] [9], pero sobre todo utilizando elementos bastante profesionales, pensados y adaptados a este tipo de desarrollos.

Este proyecto viene motivado por la falta de investigación de tipo experimental en torno a redes mesh, con dispositivos de bajo coste y que fueron diseñados para unas funciones más generalistas.

Por otra parte, en la sociedad actual existe la necesidad de estar conectado en cualquier parte y de manera permanente a Internet, o a redes de trabajo. Creemos necesario por tanto abrir un nuevo foco de investigación que ayude satisfacer las necesidades que este tipo de tecnología puede cubrir.

Por todo ello, con este proyecto se pretende diseñar y construir una plataforma de pruebas que permita realizar todo tipo de experimentaciones en redes malladas basadas en tecnologías 802.11. Aprovecharemos el espacio existente entre el suelo y el falso suelo de un laboratorio de la universidad para desplegar esta red de pruebas, con lo que tendremos la red en un entorno bastante reducido y controlado.

Además, en una segunda parte del proyecto, realizaremos pruebas de rendimiento con dispositivos 802.11 para medir las prestaciones de dicha plataforma, y caracterizar de alguna manera esta red, comprobando sus limitaciones y puntos fuertes.

1.2 Objetivos del Proyecto

El objetivo fundamental de este proyecto es desplegar una plataforma de pruebas con dispositivos comerciales de bajo coste, en un entorno reducido y controlado. Una vez construida, vamos a estudiar las prestaciones, y su rendimiento, fijándonos principalmente en un parámetro que será el volumen de información que fluye a través de la red, o throughput.

Este estudio nos servirá para comprobar como de factible sería dotar de cobertura inalámbrica áreas no demasiado extensas, como por ejemplo urbanizaciones de vecinos, centros comerciales o complejos de oficinas.

Utilizaremos enrutadores reales de uso doméstico, concretamente Linksys WRT54GL, AsusWL-500g, y Fonera 2100. Estos modelos los podemos encontrar en cualquier establecimiento de informática a un coste no muy elevado. A los dispositivos se les cambiará el firmware original para un mayor control de los mismos.

En el futuro esta plataforma de pruebas permitirá realizar diversos experimentos, como por ejemplo, el estudio de varios protocolos de enrutado.

1.3 Estructura de la memoria

Para facilitar la lectura de la memoria, a continuación se incluye un breve resumen de cada capítulo.

Capítulo 1: Introducción: En este capítulo describimos lo que pretendemos realizar en este proyecto fin de carrera. Las motivación que nos ha llevado a realizarlo y los objetivos que perseguimos a lo largo del mismo, así como los medios utilizados para su consecución.

Capítulo 2: Estado del arte: Aquí se habla de manera introductoria de las tecnologías sobre las que se centra el trabajo. Además se mencionan otros trabajos relacionados con este mismo proyecto y de los que nos hemos servido para su realización.

Capítulo 3: Despliegue de la red de pruebas: Lo que realizamos en este capítulo es una descripción de la plataforma que construimos para conseguir los objetivos del proyecto. Hablamos de los requisitos que debe tener la red, los elementos y herramientas que son necesarios para su construcción, además de las limitaciones y problemas que nos encontramos al diseñarla.

Capítulo 4: Evaluación experimental: El fin principal de este capítulo es caracterizar la plataforma de pruebas previamente construida. Realizamos distintos estudios experimentales con los que comprobamos las limitaciones y ventajas de la red. Además los experimentos nos han servido para sacar algunas conclusiones sobre redes inalámbricas malladas, ya que la última de las pruebas trata precisamente este tema de una manera más profunda.

Conclusiones, Anexos y Presupuestos: Los últimos apartados del proyecto tratan sobre las conclusiones obtenidas de trabajo realizado y del problema que nos proponemos atender. En los anexos se incluye información técnica suficiente para reproducir todo el proyecto paso a paso, manuales de las herramientas utilizadas, mapas del laboratorio, así como las modificaciones que han sido realizadas a los dispositivos de la red. Por último el presupuesto incluye un diagrama de Gantt donde están descritas las tareas para la realización del proyecto, además de un desglose de los costes atribuidos al proyecto.

Parte II

Estado del Arte

Capítulo 2

Redes Inalámbricas

2.1 Comunicaciones Inalámbricas

Una comunicación inalámbrica, como su propio nombre indica, es aquella que se realiza sin cables, utiliza el espacio como medio de propagación. En general, la tecnología inalámbrica utiliza ondas de radiofrecuencia de baja potencia y una banda de uso libre o privada para hacer posible la comunicación entre elementos.

Las redes inalámbricas fueron concebidas bajo esta idea de comunicaciones sin cables, por lo que nos podemos hacer una idea de las ventajas que presentan este tipo de redes. Por ejemplo, la movilidad, la rápida instalación de la red y por tanto los menores costes de instalación en comparación con una red convencional.

Una red inalámbrica facilita la conexión de dispositivos, mediante ondas de radio que permiten que los dispositivos móviles de un área determinada se conecten y comuniquen entre sí.

2.2 Modos de operación de redes inalámbricas

Las redes inalámbricas 802.11 pueden realizarse con o sin punto de acceso , esto es lo que determina si una red esta operando en modo “Ad-Hoc” o “Infraestructura”.

2.2.1 Red Ad-Hoc

El modo Ad-Hoc, también es conocido como punto a punto, es un método pensado para que los clientes inalámbricos puedan realizar una comunicación directa entre sí. De esta manera, no es necesario involucrar en la comunicación, un punto de acceso central. Todos los nodos de una red Ad-Hoc se pueden comunicar directamente con otros clientes. Cada uno de los nodos de la red Ad-Hoc debe configurar su adaptador en este modo, además de usar los mismos identificadores de red inalámbrica, y el mismo número de canal.

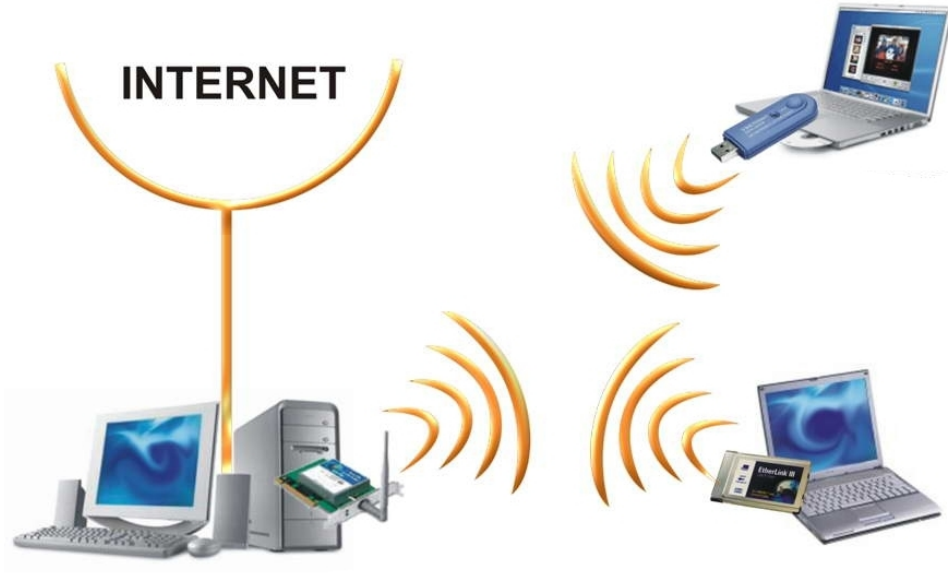


Figura 2.1: Red Ad Hoc (Fuente: http://www.debahia.com)

Si un nodo esta conectado a una red, como por ejemplo Internet, puede extender dicha conexión a otros que se conectan a él inalámbricamente en el modo Ad-Hoc.

2.2.2 Infraestructura

Al contrario que en el modo Ad-Hoc, este modo de infraestructura dispone de un elemento de coordinación central, que puede ser un punto de acceso (AP), o una estación base.

Si el punto de acceso se conecta a una red Ethernet cableada, los clientes inalámbricos pueden acceder a la red fija a través del punto de acceso inalámbrico. Además es posible conectar varios puntos de acceso entre sí, para ello deben configurarse con el mismo identificador de red, y para asegurar que se maximice la capacidad total de la red, conviene no configurar el mismo canal en los puntos de acceso que se encuentren en la misma área física de cobertura.

Los clientes descubrirán (a través del escaneo de la red) cuál canal está usando el punto de acceso de manera que no se requiere que ellos conozcan de antemano este dato.

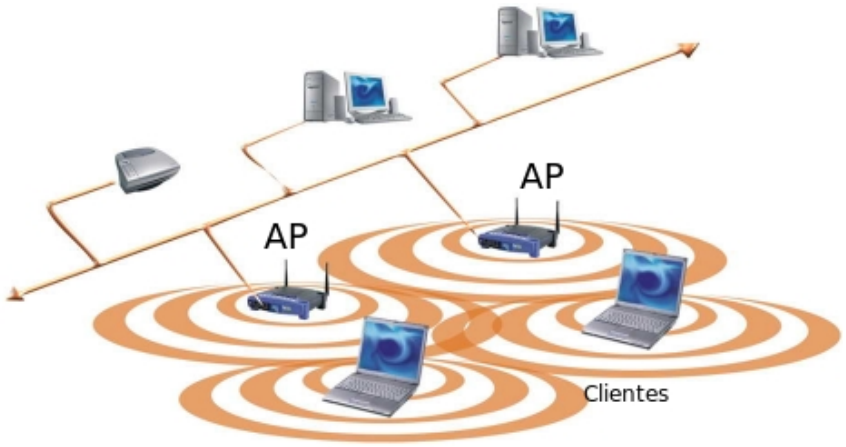


Figura 2.2: Red Infraestructura(Fuente: http://www.debahia.com)

Cuando un usuario itinerante va desde un área de cobertura de un punto de acceso a otro, el adaptador de la red inalámbrica de su equipo puede cambiar de punto de acceso, según la calidad de la señal que reciba desde distintos puntos de acceso. Los puntos de acceso se comunican entre sí a través de un sistema de distribución con el fin de intercambiar información sobre las estaciones y, si es necesario, para transmitir datos desde estaciones móviles. Esta característica que permite a las estaciones moverse sin cortes de un punto de acceso al otro se denomina itinerancia.

2.2.3 Red inalámbrica Mesh

La red inalámbrica Mesh esta basada en la topología de una red mallada multisalto inalámbrica.

Permiten que los dispositivos se comuniquen entre sí, independientemente del punto de acceso. Esto es que un cliente perteneciente a la red no tiene porque estar conectado directamente a la puerta de enlace que le da acceso a la infraestructura o al punto destino, puede estar a varios saltos del destino, en este caso la red se encargaría de enrutar a través de los distintos multisaltos inalámbricos.

Ventajas:

- Menor coste: cada nodo puede actuar como cliente y como repetidor de la red, lo que suple la necesidad de infraestructuras de repetición o nodos centrales.
- Robustez: al ser una red mallada, si uno de los nodos pierde servicio, se reduce la posibilidad de que esto afecte al resto, ya que puede existir redundancia en el camino a este nodo.
- Instalación: La complejidad en tarea de instalación de un punto mesh queda reducida, al compararlo con una red cableada, ya que simplemente habría que colocar el nodo con el software Mesh preinstalado. Al disponer de rutas dinámicas, cuando este nodo encuentre un nodo vecino, estaría dispuesto para entrar en servicio.
- Alimentación: Los nodos de la red Mesh, pueden ser construidos con requerimientos energéticos realmente bajos, por lo que pueden ser desplegados con unidades autónomas de energía, como por ejemplo solar o eólicas.

Desventajas:

- Latencia: Este tipo de tecnología puede no ser siempre buena, debido al número de saltos que puede llegar a dar un paquete hasta su destino, la red introduce un retardo que no siempre es bueno si queremos utilizar

la red por ejemplo para servicios que se requieran en tiempo real, como por ejemplo la telefonía IP.

- Compartiendo el medio: Debido al limitado número de frecuencias en que se mueven las redes WLAN actuales, puede existir interferencias entre usuarios que compartan un mismo área de cobertura física.

Soluciones:

- Para solventar este problema se puede contar con un protocolo de enrutamiento que permita transferir la información hasta su destino con el mínimo número de saltos o un número de saltos suficientemente bueno como para no perder calidad en el servicio que se quiere prestar.
- Además la utilización de estándares como el 802.11a que utilizan la banda de los 5 GHz permiten acceder a la red a más usuarios, gracias al mayor número de canales.

En la figura 2.3 podemos ver un sencillo ejemplo de aplicación de redes Mesh. Vemos como con una única línea ADSL, podemos abastecer a un área bastante extensa compuesta por 7 nodos visibles entre ellos.

Comparando tres tipos de redes Mesh inalámbricas

La arquitectura de las redes mesh inalámbricas pueden ser clasificadas en tres grandes grupos principales, basados según la funcionalidad de los nodos de la red. [1]

Red inalámbrica mesh con infraestructura. Esta topología de red jerárquica, se compone de clientes mesh, enrutadores mesh y puertas de enlace. Los router mesh constituyen la red troncal a la que se conectan los clientes. Además puede existir una o varias puertas de enlaces que sirvan para proveer acceso a Internet a la red a través de los propios routers.

Los clientes convencionales mesh pueden comunicarse directamente mediante la tecnología radio con los routers de la red. Si la tecnología radio usada

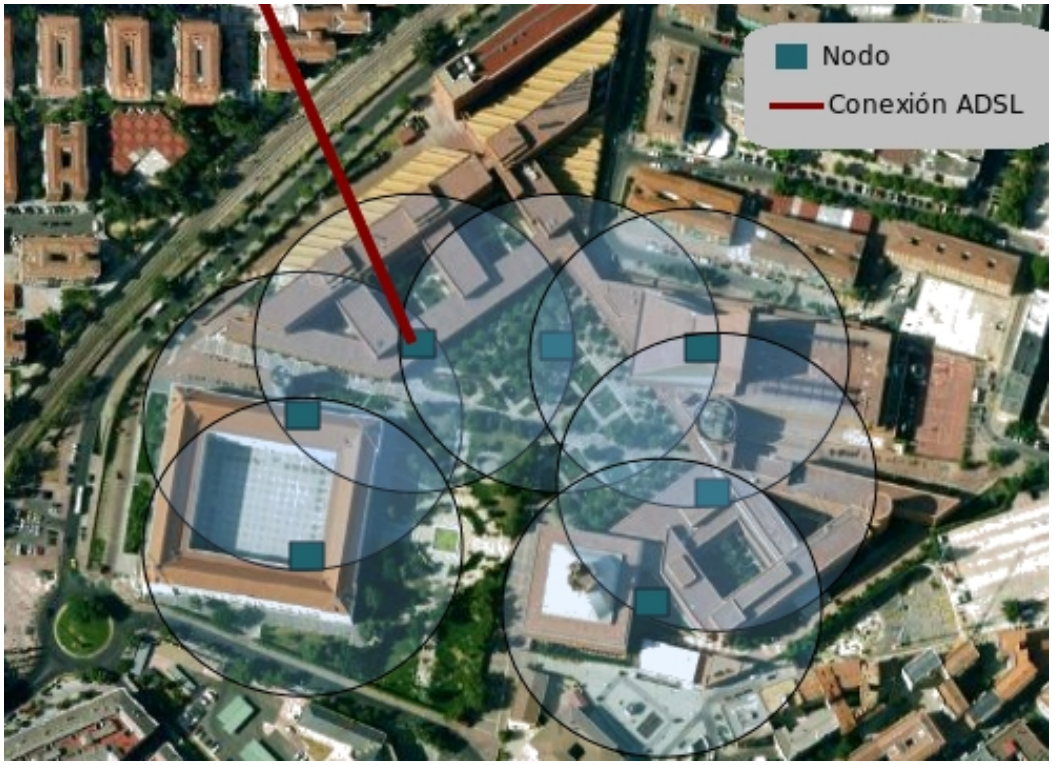


Figura 2.3: Ejemplo red Mesh

por otros los clientes fuera distinta, estos se comunicarán a través de sus estaciones base, que a su vez dispondrán de conexiones Ethernet con los routers mesh.

Red inalámbrica mesh de clientes. En este caso se prescinde de enrutadores o puertas de enlaces y las comunicaciones se producen entre los nodos clientes de la red. Los nodos clientes se encargan de enrutar y configurar la red dependiendo de las funcionalidades que hayan sido previamente definidas por los usuarios.

Un ejemplo de esta arquitectura sería el de la figura 2.5, en esta red un paquete destinado a un cierto cliente, puede saltar a través de los distintos nodos hasta llegar a su destino. Los nodos clientes normalmente conforman la red mesh usando el mismo tipo de dispositivos radio.

Red inalámbrica mesh híbrida. Esta arquitectura de red, que podemos ver en la figura 2.6, combina los dos tipos de redes comentadas anteriormente. Los clientes pueden acceder a la red, tanto a través de los routers mesh como utilizando otros nodos clientes intermediarios, que a su vez proveerán

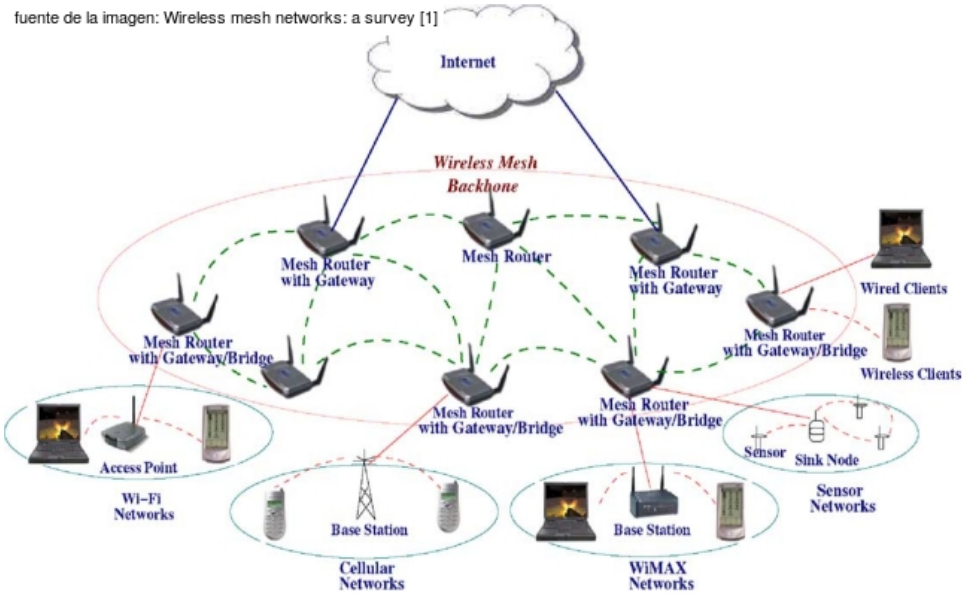


Figura 2.4: Red inalámbrica mesh con infraestructura

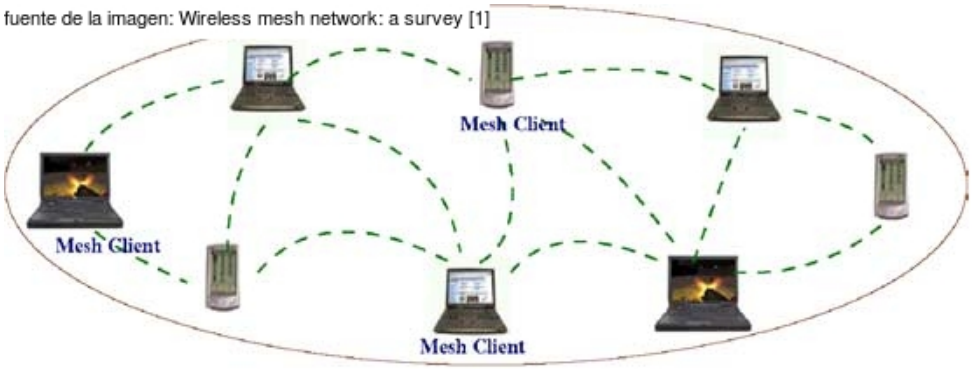


Figura 2.5: Red inalámbrica mesh con clientes

de conectividad a Internet o a otras redes de trabajo a los clientes finales.

2.3 Despliegues de redes Mesh

Los inicios de las redes mesh son, como no, militares. Inicialmente se usaron para comunicarse con aquellas unidades de militares que aún estando lejos de las zonas de cobertura de sus mandos estaban lo suficientemente cerca entre si como para formar una cadena a través de la cual se pudiese ir pasando los mensajes hasta llegar a su destino (los mandos).

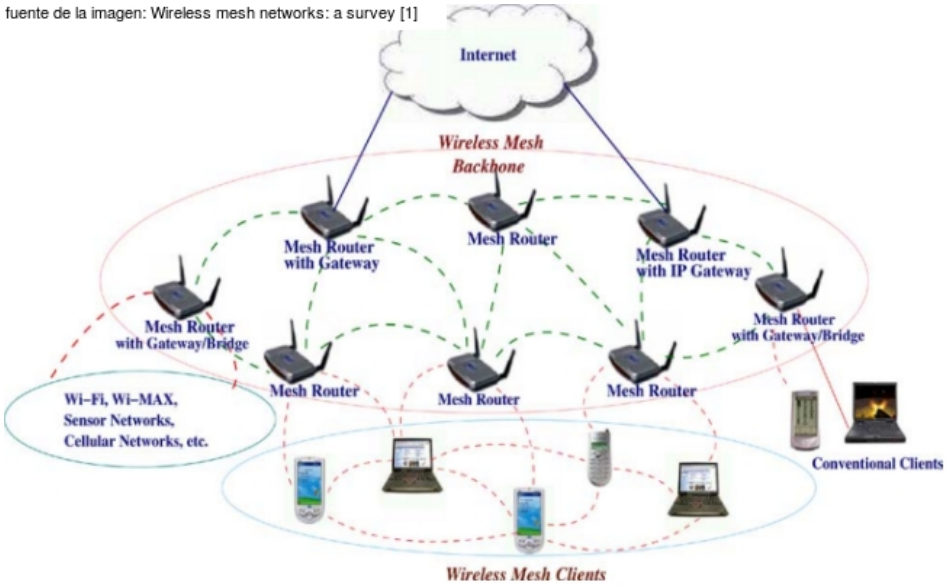


Figura 2.6: Red inalámbrica mesh híbrida

Antiguamente no se usaban tanto porque el cableado necesario para establecer la conexión entre todos los nodos era imposible de instalar y de mantener. Hoy en día con la aparición de las redes inalámbricas este problema desaparece y nos permite disfrutar de sus grandes posibilidades y beneficios.

Actualmente este tipo de redes esta bastante extendido en toda clase de grupos. Empresas privadas dedicadas a la comercialización de redes mesh, instituciones sociales o comunitarias, o incluso laboratorios y grupos de investigación dedicados al estudio estas redes y todo lo que las rodea.

2.3.1 Redes mesh comunitarias

Las redes mesh comunitarias están formadas por agrupaciones de usuarios, instituciones o empresas que deciden construir una red mesh, y de esta manera conectarse entre ellos con altas prestaciones y un bajo coste, además de servicios de valor añadido.

El objetivo de estas comunidades de usuarios no es solamente de posibilitar el acceso a Internet. Es mucho más ambicioso. Se trata de crear otra red, pero gestionada por sus propios usuarios. También pretenden acercar la tecnología a la sociedad, crear nuevos canales gratuitos de comunicación entre las

personas e, incluso, ser una red de soporte alternativa en caso de catástrofe.

Un ejemplo de este tipo de redes a nivel nacional es Guifi.net [1] en la que ya existen casi 13000 nodos conformando la red mesh, de los cuales hay unos 9500 operativos a día de hoy.

Y también en otros países como argentina tienen sus redes comunitarias como la de Lugro Mesh [2] en la que se investiga sobre redes Wireless utilizando Software Libre.

También existen proyectos como los de la fundación EHAS[3] donde utilizan las redes mesh para mejorar los procesos de salud en zonas rurales aisladas de países en desarrollo. Esta asociación dan soporte a redes de telemedicina rural en zonas aisladas, y realizan proyectos como el de un router solar autoconfigurable para redes Mesh [6] .

Un beneficio que nos puede ofrecer este tipo de red comunitaria es el de disponer de una red de emergencia para su uso en caso de catástrofe: En caso de catástrofe y el consiguiente colapso de las redes de comunicación habituales la red mesh será una alternativa de comunicación al no depender de los canales, medios de transmisión habituales permitiendo conectar a la red desde cualquier punto y en todo momento para servir de red de emergencia y atender a las necesidades de comunicación y transmisión de voz y datos que puedan surgir.

2.3.2 Redes mesh comerciales

También hay algunas empresas que se dedican a desplegar redes inalámbricas a todo tipo de clientes, tanto particulares, como profesionales. Nodalis[4] es una empresa que se dedica a ello, y presenta soluciones de interconexión de redes de un modo más profesional que las redes comunitarias. Llegando a ofrecer servicios como la monitorización de la red, supervisión online o gestionar puntos desde los que el cliente puede cobrar por dar uso de la red.

[1] http://guifi.net/es
[2] http://www.lugro-mesh.org.ar
[3] http://www.ehas.org
[4] http://www.nodalis.es

Incluso hay dispositivos en el mercado que están pensados específicamente para la implantación de este tipo de redes y con los que se facilita la tarea e instalación de la misma, como por ejemplo los de la empresa Meraki.[5]. Los dispositivos que comercializa esta empresa tienen la peculiaridad de que son muy económicos, fáciles de instalar, e incorporan enrutamiento avanzado: cada nodo transmite automáticamente entre sí, formando redundancias automáticas a través de múltiples trayectorias, aumentando alcance y eficiencia. Además es self-healing, es decir, la red se arregla y se reconfigura a sí misma si algún nodo está fuera de servicio, lo que disminuye la necesidad de mantenimiento.

Plettac Electronics es una empresa dedicada, entre otras actividades, a la implantación de redes inalámbricas malladas en exteriores. Esta empresa ha realizado diversos despliegues de redes mesh en algunos barrios de ciudades como Madrid, y lo ha hecho aprovechando el mobiliario urbano como soporte para su instalación (farolas, semáforos,etc.). Estas redes pueden soportar servicios esenciales como la comunicación con la policía, bomberos, servicios sanitarios o información de tráfico en áreas metropolitanas.

Figura 2.7: Aplicación práctica de red mesh en área metropolitana

2.3.3 Redes mesh de laboratorio

Una de las redes mesh de laboratorio más interesantes es Roofnet [6], se trata de una red 802.11b/g experimental desarrollada por el MIT [7], en la que actualmente hay unos 20 nodos activos distribuidos por la ciudad de Cambridge.

[5] http://meraki.com
[6] http://pdos.csail.mit.edu/roofnet/doku.php
[7] Instituto de Tecnología de Massachusetts

Inicialmente el banco de pruebas de este proyecto comenzó siendo una plataforma interna, parecida a la que hemos desarrollado en nuestro trabajo. Posteriormente la red de pruebas se modificó y pasó a ser una red exterior, en la que con sus más de 20 nodos activos actualmente cubren un área de cobertura de aproximadamente seis kilómetros cuadrados.

La red es parte de un proyecto de investigación en el que se pretende, realizar mediciones a nivel de enlace de 802.11 [9], encontrar rutas de alto rendimiento para soportar aplicaciones susceptibles a latencia, adaptación de enlaces [8], y desarrollar nuevos protocolos de comunicación que aprovechen mejor las características de las comunicaciones radio.

Uno de los protocolos de comunicaciones que ha surgido de RoofNet es B.A.T.M.A.N.[8] se trata de un protocolo de encaminamiento desarrollado específicamente para este tipo de redes descentralizadas en las que las rutas son dinámicas y pueden cambiar constantemente.

Otro proyecto relacionado con redes mesh, a nivel europeo, es CARMEN (CARrier grade MEsh Networks)[9], en este proyecto intervienen varios socios, entre los que destacan la universidad Carlos III de Madrid, la universidad AGH de Cracovia, British Telecomunications PLC, o Deutsche Telekom AG.

El proyecto CARMEN estudia y especifica una red de malla inalámbrica que permita servicios de portador grado triple-juego (voz, vídeo y datos) para operadores de redes móviles y fijas. Los futuros operadores de redes se compondrán de un núcleo de red común y varias redes de acceso, y la red de acceso CARMEN complementará otras tecnologías de acceso proveyendo tecnología de acceso de red de malla de rápido despliegue y de bajo coste.

Este proyecto fin de carrera se enmarca dentro de lo que es el proyecto europeo CARMEN. Parte de los resultados obtenidos aquí se han utilizado para este proyecto, además la plataforma desplegada se utilizará en el futuro para evaluar el rendimiento de parte de los componentes desarrollados en el proyecto CARMEN.

[8] Better Approach To Mobile Adhoc Networking,

[9] http://www.ict-carmen.eu

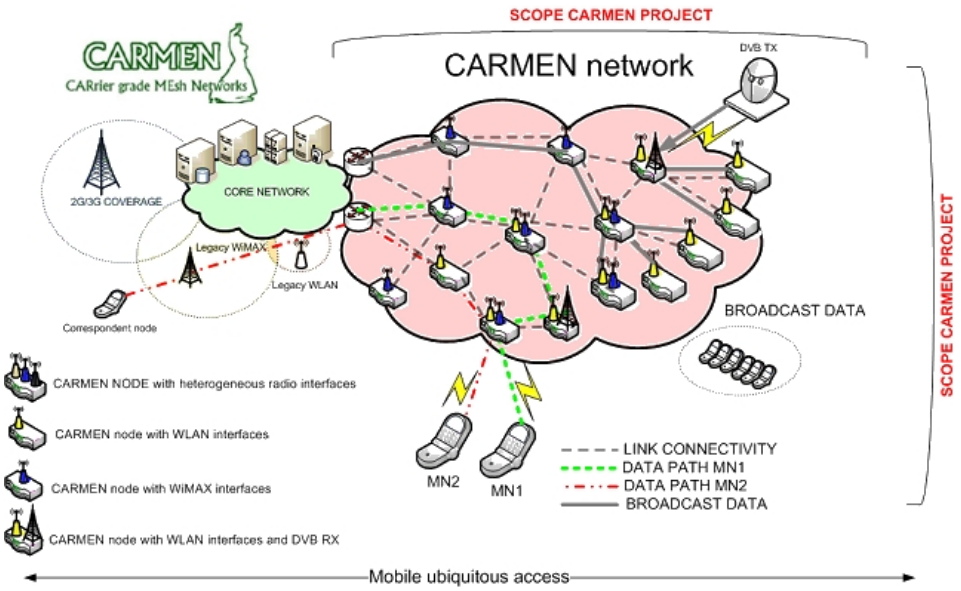


Figura 2.8: Topología de la red CARMEN

2.4 Estándares IEEE 802.11

El estándar 802.11 viene definido por el IEEE[10] especificando sus normas de uso en una red WLAN. En este proyecto se describe a continuación, el uso de algunos estándares 802.11, denominando con sufijos (a/b/g) los cuales son revisiones, complementos y mejoras de este protocolo 802.11.

2.4.1 IEEE 802.11a

Se trata de una extensión del 802.11. Provee una tasa de hasta 54 Mbps operando en la banda de 5 GHz utilizando 52 subportadoras con multiplexación por división en frecuencias ortogonales. A pesar de la velocidad que ofrece los equipos que trabajan con este estándar no pueden penetrar tan lejos como los del estándar 802.11b/g dado que sus ondas son más fácilmente absorbidas.

Esta tecnología de velocidad mayor que 802.11b permite que las redes locales inalámbricas tengan un mejor rendimiento en aplicaciones multimedia. No puede interoperar con equipos del estándar 802.11b, excepto si se dispone

[10]http://www.ieee.org

de equipos que implementen ambos estándares.

Canales de 802.11a

El utilizar la banda de 5 GHz representa una ventaja del estándar 802.11a, dado que se presentan menos interferencias.

Los identificadores de canales y frecuencias centrales, para cada canal usado por IEEE 802.11a son los que se muestran en la siguiente imagen:

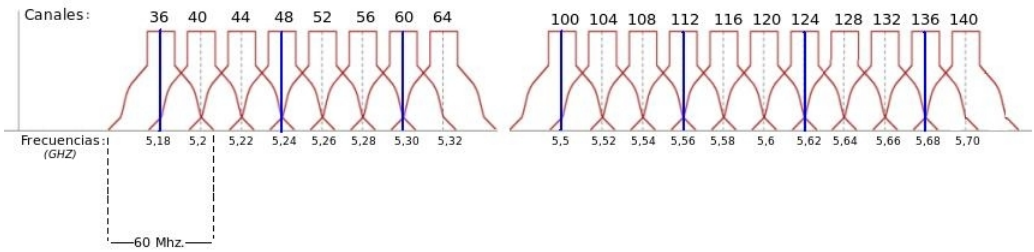


Figura 2.9: Canales en 802.11a

2.4.2 IEEE 802.11b

El estándar 802.11b funciona en la banda de los 2.4GHz y tiene una velocidad teórica máxima de 11Mbps. Utiliza el método de acceso múltiple por detección de portadora (CSMA/CA).

2.4.3 IEEE 802.11g

Este estándar se considera la evolución del 802.11b. Utiliza la misma banda que su predecesor, la de 2.4GHz.

802.11g consigue operar a una velocidad teórica máxima de 54Mbps gracias al uso de la misma tecnología de modulación que el 802.11a, multiplexación por división en frecuencias ortogonales, pero en la banda de 2.4GHz.

Canales de 802.11b/g

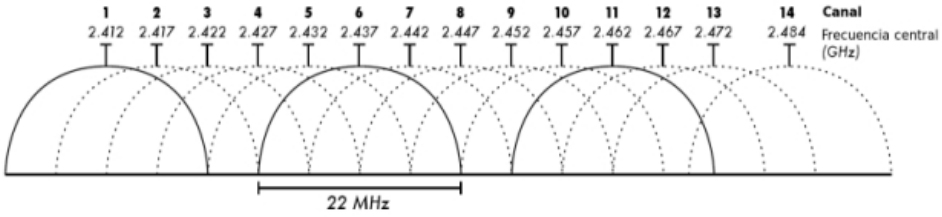


Figura 2.10: Canales en 802.11b/g

2.4.4 IEEE 802.11n

Hace menos de un año se aprobó este nuevo estándar que ofrece mucha mayor velocidad y mejor rendimiento, es posible llegar a una velocidad real de transmisión de 600Mbps, y el alcance de operación supera a sus predecesores gracias a la tecnología MIMO[11], que permite utilizar varios canales a la vez para enviar y recibir datos gracias a la incorporación de varias antenas.

A diferencia de las otras versiones de Wi-Fi, 802.11n puede trabajar en dos bandas de frecuencias: 2,4 GHz (la que emplean 802.11b y 802.11g) y 5 GHz (la que usa 802.11a). Gracias a ello, 802.11n es compatible con dispositivos basados en los modos anteriormente comentados.

2.4.5 IEEE 802.11s: Mesh

Define la interoperabilidad de fabricantes en cuanto a protocolos Mesh. A día de hoy no existe un estándar como tal, por eso cada fabricante tiene sus propios mecanismos de generación de redes malladas.

[11]Multiple Input - Multiple Output

Desde hace tiempo algunos fabricantes como Intel, han realizado propuestas[12] para el estándar 802.11s. Desde este fabricante se propone un estándar compatible con los existentes 802.11a/b/g, que permite la creación de mallas.

2.5 Conclusiones

Como hemos visto en esta parte hay una gran variedad campos abiertos en este tema de redes malladas multisalto, sobre todo para entornos exteriores.

Disponemos de estándares desarrollados por organismos internacionales, protocolos de comunicaciones, redes comerciales, incluso hay dispositivos que están pensados para instalar de una manera profesional redes mesh. Lo que no hemos encontrado son muchos estudios sobre redes mesh desarrolladas con dispositivos de bajo coste y accesibles por cualquier particular.

Aprovechando la tecnología existente, basándonos en la documentación estudiada sobre este tema y con ayuda de algunas herramientas, en el siguiente punto diseñaremos y desarrollaremos una plataforma que nos permita realizar experimentos y testeos de una red mallada inalámbrica con elementos de bajo coste.

[12] http://www.zdnet.co.uk/news/networking/2005/03/03/intel-hangs-mesh-hopes-on-80211s-39189953

Parte III

Trabajo realizado

Capítulo 3

Despliegue de la Red de Pruebas

En este capítulo vamos a hablar de como se realizó el despliegue de la plataforma de pruebas, así como los problemas que nos fueron surgiendo durante la implantación de la misma. Primero se pensó en los requisitos que debería tener, como por ejemplo el poder ser gestionada de manera remota, gracias a lo cual no será necesario desplazarse hasta el laboratorio para realizar algún cambio no físico en la red.

Pretendemos también que la red de pruebas se encuentre en un entorno interior, controlado y reducido, por lo que se decidimos aprovechar el espacio existente entre el suelo y el falso suelo, veremos las ventajas e inconvenientes que tiene el desplegar la red de esta manera.

También describiremos los dispositivos utilizados, la configuración que tendrá cada uno de ellos, así como las herramientas utilizadas para gestionar y llevar a cabo pruebas en nuestra plataforma.

3.1 Diseño de la red

3.1.1 Requisitos

Como requisito para este proyecto fin de carrera, se pide construir una red de comunicaciones con elementos de bajo coste, y que sean de fácil adquisición.

El hecho de tener que desplazarse para realizar algún cambio en la red hay que tenerlo en cuenta a la hora de diseñar la red de pruebas, por ello necesitamos que los elementos de sean gestionados de manera remota, y a ser posible por un sistema operativo GNU/Linux. Una vez tengamos desplegada la plataforma no será necesario estar físicamente en el laboratorio para realizar las pruebas de rendimiento. El único inconveniente es a la hora de resetear los routers, esta tarea hay que realizarla a mano si el router se queda bloqueado, por lo que no hay más remedio que desplazarse hasta el laboratorio.

Entre los distintos elementos, se necesitarán al menos dos equipos para monitorización y gestión de la red. Estos equipos deberán ser lo suficientemente potentes como para soportar varias horas o incluso días realizando cálculos para las pruebas. Como vamos a realizar pruebas con redes inalámbricas deberán tener instalado dispositivos de red inalámbricas. Además deberán tener una tarjeta de red LAN, ya que la gestión y monitorización de los dispositivos se realiza mediante la red cableada.

El tener los nodos conectados a los PCs de control a través de la red cableada nos permite separar en cierta medida la red inalámbrica, de la cableada ya que de no ser así, podría influir en la toma de datos durante la realización de las pruebas y testeos de la red.

Dadas las limitaciones de espacio que tiene el laboratorio debemos instalar un número de nodos tal que se encuentren separados una distancia aceptable y a la vez sean los suficientes como para realizar pruebas con un número considerable de saltos.

3.1.2 Descripción del Entorno

La red se despliega bajo el suelo del laboratorio 4.1.F.04 del edificio Torres Quevedo de la universidad Carlos III de Madrid. Este laboratorio se encuentra en el Departamento de Telemática, y reúne una serie de condiciones que le hacen óptimo para ubicar en él la red que hemos diseñado para los experimentos.

El laboratorio dispone de equipos de comunicaciones de toda clase, tales como concentradores, conmutadores, enrutadores, además de la instalación

eléctrica oportuna que nos facilita la tarea de alimentar los equipos. Las condiciones de temperatura y humedad también son adecuadas a los equipos ya que el laboratorio se encuentra aclimatado, gracias a esto las condiciones externas de calor o humedad no interfieren en los experimentos, cosa que si que habría que tener en cuenta si la instalación se realizase en el exterior.

El falso suelo de este laboratorio se compone de losetas de madera y de una pequeña lámina metálica de 5mm.de grosor. Esta lámina metálica nos ofrece un aislamiento, gracias a lo cual tenemos la red aislada en cierta medida del exterior. En el apartado 4.2, estudiamos más a fondo como afecta a las prestaciones y al rendimiento de la red este fenómeno, realizando algunos experimentos.

3.1.3 Descripción de la red

El despliegue de la red forma parte de un despliegue progresivo que se está llevando a cabo dentro del departamento de Ingeniería Telemática. Se han realizado ya experimentos [5] en los que se aprendió cuales eran los mejores dispositivos y cada cuánto espacio hay que desplegarlos para permitir hacer experimentos multisalto jugando con la potencia.

Teniendo en cuenta las limitaciones de espacio en el laboratorio descrito anteriormente la red constará de 14 nodos principales, los cuales se situarán perfectamente equiespaciados unos de otros, como se ha diseñado en la figura 3.1. Se ha pensado esta manera de situar los nodos, con el fin de tener la mayor homogeneidad posible y evitar dar preferencia o penalizar a nodos que se encuentren mejor o peor situados.

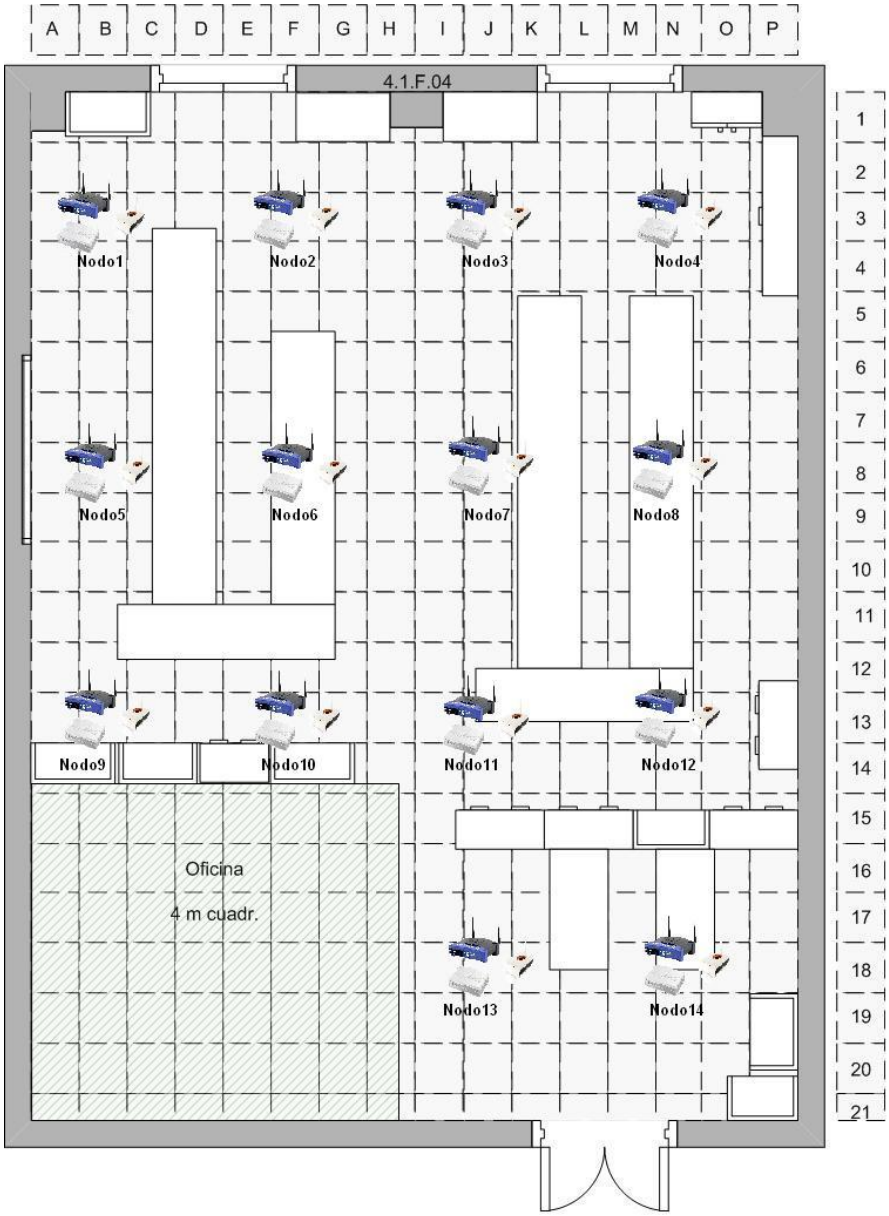


Figura 3.1: Situación de los nodos en la red de pruebas del laboratorio

Hemos conectado entre sí los nodos, por medio de dos concentradores D-LINK de 24 puertos cada uno, ubicados en el centro del laboratorio, a los que también hemos conectado dos PC de control, que además de gestionar y monitorizar la red, nos servirán para realizar las pruebas. Estos PC tienen un sistema operativo Ubuntu 7.09 y son accesibles remotamente desde Internet por ssh.

Antes de realizar este montaje final, se pensó en otras maneras de conectar los elementos de la red. De hecho se llegó a hacer una instalación, en la que cada uno de los nodos estaban interconectados a un pequeño Hub y este a su vez a los concentradores D-LINK. Desestimamos esta instalación ya que los tiempos de latencia entre los nodos a través de la red cableada eran bastante altos.

Teniendo en cuenta los requerimientos técnicos expuestos anteriormente, y basándonos en trabajos anteriores [5] se ha decidido que cada uno de los 14 nodos de la red estén compuestos por los siguientes modelos de enrutadores:

- Linksys WRT54GL v1.1: Se trata de un router inalámbrico que además incluye un conmutador interno de 5 puertos. Esta equipado con un procesador de 200Mhz, y una capacidad de memoria RAM de 16MB. El interfaz Wireless soporta las normas IEEE 802.11b/g y el conmutador la norma IEEE 802.3 Ethernet. El atractivo principal de este equipo es que, nos permite una total compatibilidad con el firmware OpenWrt, gracias a esto podemos modificar y programar muchos aspectos del router de manera sencilla.

Figura 3.2: Linksys WRT54GL

- Asus WL-500g Premium: Este router esta equipado con un procesador a 266Mhz, y una memoria RAM de 32MB. Además tiene un interfaz WLAN IEEE 802.11b/g y un conmutador interno de 5 puertos Ethernet IEEE 802.3.

  Este modelo dispone de un pequeño slot mini-PCI el cual permite cambiar la tarjeta Wireless original. Se quitó la tarjetas principal de comunicaciones (Broadcom) y se ha sustituido por otra Atheros basadas en la norma IEEE 802.11a/b/g (Alpha Networks AWPCI085S). Esta tarjeta funciona gracias a los drivers de Madwifi[1]. Gracias al cambio efectua-

[1] http://madwifi.org/

do podemos usar la banda de frecuencias de la norma 802.11a, por lo que necesitamos cambiar también la antena original (2,4 Ghz.) por una antena externa de banda ancha y baja ganancia (8 dBi), concretamente se trata de la Asus WL-Ant 168.

Figura 3.3: Asus WL-500g

- Fonera 2100: Es un router inalámbrico con un procesador a 170Mhz.y una memoria RAM de 16MB, con un firmware basado en OpenWRT. Disponen de un solo puerto Ethernet IEEE 802.3 y un dispositivo Wifi IEEE 802.11b/g. Al igual que los otros dos routers, las foneras han sido modificadas para facilitar su gestión, en los Anexos a este proyecto se describe el proceso entero de modificación de los elementos.

Figura 3.4: Fonera

Cada uno de los nodos ocupa aproximadamente el espacio físico de una loseta del falso suelo del laboratorio (50x50 cm), y para que la instalación sea lo más homogénea posible se han ubicado todos de igual manera, como podemos observar en la figura 3.6, quedando así todas las antenas orientadas de igual manera.

Se ha tomado una fotografía de uno de los nodos que componen la red para un mejor detalle (figura 3.5).

El firmware original de todos y cada uno de los routers ha sido sustituido por uno que nos permitiera un mayor control sobre el manejo de los distintos parámetros del dispositivo. Basándonos en experiencias de proyectos anteriores el firmware elegido para cada uno de los enrutadores ha sido el siguiente:

Figura 3.5: Fotografía de un nodo de la red de pruebas

Asus: Una versión de OpenWrt[2], concretamente la kamikaze 7.09 para chipsets atheros 2.6

Linksys: Al igual que en el modelo Asus usamos la versión kamikaze 7.09 de OpenWrt con kernel 2.4.

Fonera: Al no disponer de experiencia previa con este dispositivo, se decidió comenzar probando, al igual que los elementos anteriores, distintas versiones de OpenWrt. Como explicamos en el anexo B.2.1, al encontrarnos con varias incompatibilidades entre la Fonera y los distintos firmwares probados se decidió instalar finalmente una versión de otro tipo de firmware basado en DD-WRT[3], exactamente la versión DD-WRT v24 RC6.2 para tarjetas Atheros WiSoc.

Al final de este proyecto se incluyen los anexos B.1, y B.2, en los que se describe paso a paso la configuración de estos firmwares en los dispositivos Asus y Foneras, no se incluye la instalación del firmware para el modelo de

[2]http://openwrt.org/
[3]http://www.dd-wrt.com/site/index

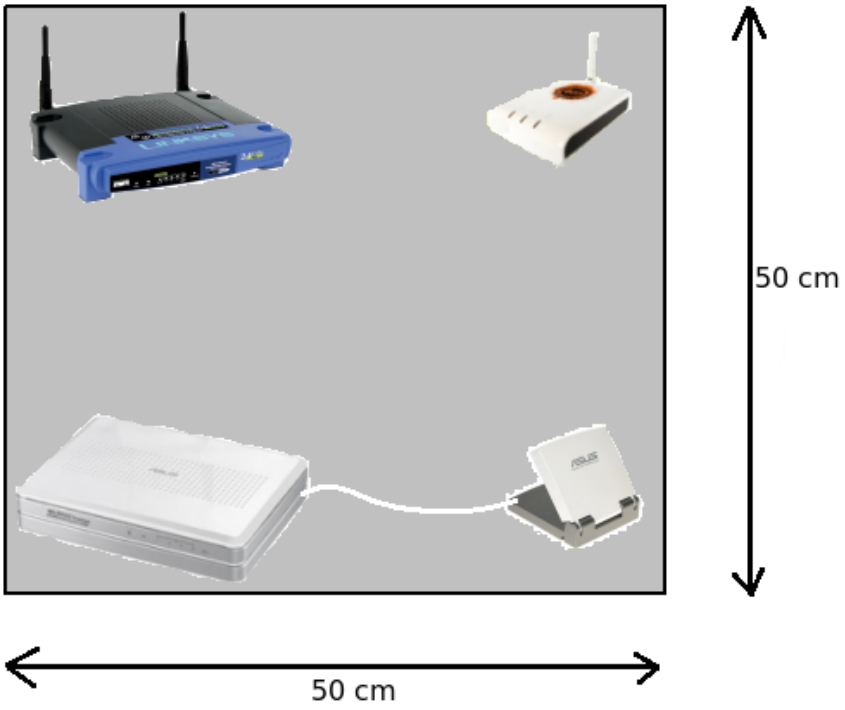


Figura 3.6: Situación de elementos bajo las losetas

router Linksys ya que no se han utilizado para las pruebas, se puede ver en otro proyecto fin de carrera [7].

3.1.4 Coste de los equipos

Una de las mayores ventajas de este tipo de redes mesh, es que con un coste no muy elevado podemos cubrir un área bastante extensa. Por lo que su despliegue puede ser una buena idea desde el punto de vista económico, y la relación calidad precio justifica su uso.

Los costes aproximados por unidad serían los mostrados en la tabla 3.1.

Despreciando el coste del cableado, los equipos de control, y los dos concentradores utilizados, el coste total de los equipos utilizados para la red asciende a 2870 euros.

Para ver en mayor detalle el presupuesto de la red podemos ir al anexo A que se adjunta al final de este proyecto.

Desglose de los Costes del equipamiento del proyecto			
Equipo	**Coste unitario(€)**	**Unidades**	**Coste Total(€)**
Linksys Wrt54GL	52	14	728
Asus WL-500 GP	75	14	1050
Antena Asus WL-ANT 168	22	14	308
Tarjeta Atheros para IEEE 802.11a	36	14	504
Fonera 2100	20	14	280
Total			**2870,00 €**

Tabla 3.1: Desglose de los Costes del equipamiento principal del proyecto.

3.2 Instalación y Configuración

Como se ha comentado antes, previo a la instalación de los equipos en el subsuelo, hay que realizar una actualización del firmware de los mismos para poder manejarlos mejor a la hora de realizar las pruebas. Una vez finalizado este proceso, comenzamos la instalación del cableado de red y eléctrico para poder dotar a cada uno de los puntos de alimentación y de una vía de comunicación cableada con los equipos de gestión.

Para el cableado de comunicaciones utilizaremos cable de red clase UTP categoría 5E según el estándar TIA/EIA-568-B[4] con conectores RJ-45.

Los equipos, serán gestionados a través de su interfaz Ethernet 802.3, los configuraremos del tal manera que se encuentren dentro de la misma subred y con el siguiente direccionamiento IP:

IP de la subred: 192.168.200.0 - Mascara: 255.255.255.0

Con esto tendremos direcciones suficientes para poner instalar hasta 254 equipos. Para llevar un orden a la hora de asignar direcciones lo haremos de la siguiente forma:

Linksys: desde 192.168.200.1 , hasta 192.168.200.14

Asus: desde 192.168.200.101 , hasta 192.168.200.114

Foneras: desde 192.168.200.201 , hasta 192.168.200.214

[4]http://www.tiaonline.org/

En los Anexos del proyecto se adjunta una tabla B.1 en la que de forma más detallada, se incluye la dirección IP de cada uno de los equipos, así como su nombre y ubicación dentro del laboratorio B.7.

Además, para facilitar la tarea de gestión de los dispositivos, se les asocia un nombre del tipo: CMPxxx[5], siendo xxx el último segmento de la dirección IP, por ejemplo el equipo con la dirección 192.168.200.8 se llamaría CMP008.

Como es obvio la tarjeta de Ethernet del PC de control tendrá una IP perteneciente a la subred antes descrita, concretamente se trata de la dirección: 192.168.200.230, que apunta al nombre: *gusano*, además de tener una IP pública para gestión remota (163.117.140.50). El otro PC de gestión (quisquilla), tiene la dirección IP local 192.168.200.231 y una dirección pública 163.117.140.119, para su acceso remoto.

En este equipo de control se instaló un repositorio local con el software específico, de manera que podíamos instalar el software necesario en los routers sin la necesidad de que estos estuvieran conectados a Internet.

En la figura 3.7 vemos como sería el esquema de la topología lógica de la red de pruebas. La elipse de color rojo representa la red cableada, vemos como están conectados los dispositivos así como la dirección IP que se les asigna a cada uno de ellos, la elipse azul representa la red inalámbrica que conformarían los dispositivos inalámbricos y el rango de direcciones IP correspondiente a cada tipo de elemento.

[5]CARMEN Mesh Point

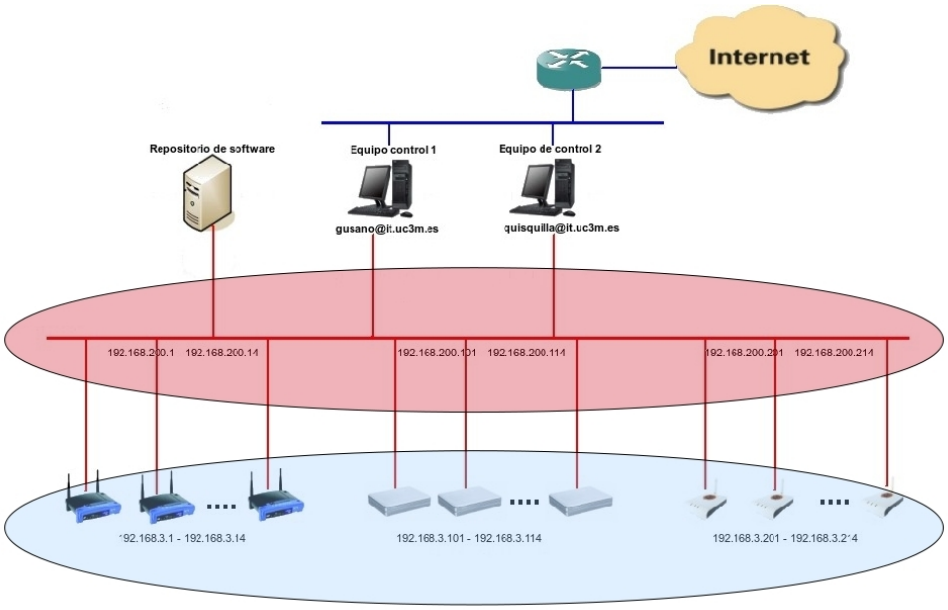


Figura 3.7: Esquema de red, topología lógica cableada e inalámbrica

Conviene destacar que conseguimos una separación entre la red cableada y la red inalámbrica rompiendo el *bridge* interior que existe en los routers. Originalmente los routers Linksys WRT54GL y Asus WL-500G disponen de 2 vlanes, una que une los puertos de la red cableada y otra para el puerto que se conecta a Internet. A su vez existe un bridge entre los puertos de la red cableada, y la red inalámbrica, puesto que vamos a utilizar la red cableada para gestionar los dispositivos y el interfaz inalámbrico para enviar y recibir datos, sería conveniente que estos quedaran separados. El proceso de como llevar a cabo esta separación esta explicado detalladamente en el anexo B.1 de este proyecto.

3.2.1 Herramientas utilizadas

Para poder llevar a cabo los experimentos, con el fin de caracterizar nuestra red, es necesario utilizar ciertas herramientas que nos faciliten esta tarea.

En Internet disponemos de un amplio catálogo de utilidades que nos ayu-

darán a realizar las pruebas. Preferentemente hemos seleccionado software que sea open source y esté bajo licencia GNU GPL[6].

A continuación describiremos brevemente el software utilizado, en el anexo B.4 de este proyecto se explica de una manera más detallada como usar este software:

iperf

Para la evaluación de rendimientos en las comunicaciones en nuestra red local y posterior optimización de los parámetros, disponemos de multitud de herramientas, una de ellas es iperf [7]. Con iperf podemos medir el ancho de banda y rendimiento de una conexión entre dos nodos. Se trata de una herramienta cliente-servidor, por tanto tendremos que ejecutar iperf en las dos máquinas, una de ellas hará de servidor y otra de cliente.

Con esta herramienta podemos generar tráfico de tipo TCP o UDP, con el ancho de banda que elijamos, y enviarlo a través de la red durante el tiempo que consideremos necesario.

nagios

Nagios[8] se trata de un sistema de código libre de monitorización de redes, con el vigilamos los equipos de la red, y los servicios de la misma. Además nos alerta cuando el comportamiento de la red no es el que deseamos. Entre los servicios de red que podemos monitorizar destacan el tráfico HTTP, STMP, SNMP, etc... En cuanto a la monitorización de los sistemas hardware podemos monitorizar el estado de los puertos, la carga de procesador, o el uso de la memoria.

[6] http://www.gnu.org/licenses/licenses.es.html
[7] http://iperf.sourceforge.net/
[8] http://www.nagios.org/

tcpdump

Gracias a la herramienta tcpdump[9] podemos analizar el tráfico que circula por la red. Nos permite capturar y mostrar en tiempo real los paquetes transmitidos y recibidos en la red a la que el equipo monitor se encuentra conectado. Para este proyecto, hemos configurado el tcpdump de modo que escuche en la interfaz inalámbrica y utilizando ciertos filtros nos quedamos con los paquetes que nos interesan.

3.3 Conclusiones

La red que se ha diseñado podría ser empleada para realizar varios tipos de experimentos, como por ejemplo probar protocolos de comunicaciones de voz sobre ip, videoconferencia, juegos en tiempo real, u otras aplicaciones que requieran una transmisión de datos por la red los cuales tengan tiempos de latencia bajos.

Consideramos que estos experimentos son algo avanzados, y previamente necesitamos tener algo más definida la capacidad general de la red. Para esto, en la siguiente parte del proyecto, comprobaremos si la red que hemos realizado es más o menos consistente en cuanto a rendimiento de la misma, realizaremos para ello una serie de experimentos que nos permitirán conocer algo mejor nuestra plataforma de pruebas.

Además responderemos a una de las cuestiones del proyecto, ya que con estos experimentos podemos decir si este tipo de red, contruida con elementos de bajo coste, es apta o no para desplegar redes multisalto inalámbricas.

[9] http://www.tcpdump.org/

Capítulo 4

Evaluación Experimental

Con el fin de caracterizar la red y que quede de la manera más homogénea posible dentro del entorno en el que ha sido desplegada, vamos a comenzar nuestro estudio realizando unos experimentos para comprobar cual sería el entorno ideal a la hora de realizar las pruebas.

Estudiaremos si el efecto de tener la red aislada en el subsuelo resulta beneficioso o si es irrelevante para el rendimiento de la red. Además realizaremos un experimento que nos dirá si es mejor o no el hecho de generar el tráfico de las pruebas en los PCs o directamente en los routers. También haremos un estudio sobre cual es el mejor horario para la ejecución de las pruebas y ver si ciertos canales resultan más beneficiosos que otros para el rendimiento de nuestra red.

Haremos un estudio de cómo interfieren entre sí los distintos enlaces que pueden estar funcionando a la vez en la plataforma de pruebas, además de cómo afecta el variar la potencia en estos enlaces.

Por último configuraremos la topología de nuestra red de pruebas para conseguir una red mallada, en la que correremos un algoritmo heurístico con el que conseguiremos averiguar cual sería la mejor parametrización posible de la red mallada.

Hay que destacar que en este proyecto nos centraremos en realizar pruebas para los modelos de routers Asus WL-500g y Fonera 2100, funcionando en modo 802.11g. El modelo Linksys WRT54GL se estudió en otro proyecto

fin de carrera [7], y el modo 802.11a de los Asus fue estudiado en otro trabajo previo [5].

4.1 Diferencias entre generar trafico en PC y routers

En el despliegue que se ha descrito en el apartado 3.1.3, hablábamos de la utilización de dos PCs para realizar las pruebas de rendimiento de la red. Otra posible configuracion sería prescindiendo de estos dos ordenadores y utilizar los propios routers inalámbricos para generar y enviar el tráfico en la red, lo cual nos ahorraría dos equipos de gestión y monitorización.

Hemos realizado un experimento para comprobar si el hecho de generar tráfico en los dispositivos inalámbricos o en los PCs afecta al rendimiento de la red, y si es posible prescindir de los PCs, desde el punto de vista de la optimización.

Para esta prueba hemos utilizado la herramienta *iperf*, con la que hemos generado e inyectando tráfico UDP durante 30 segundos y con una tasa de envío de 35Mbps. Hemos realizado 5 mediciones para varios tamaños de trama distintos, comenzando en 100 bytes, con un paso de 100 bytes y terminando en 1500.

En primer lugar utilizamos uno de los dos PCs para generar y enviar el tráfico a través de la red cableada hasta uno de los dos routers, el cual a su vez envía estos datos al otro router a través de la red inalámbrica, finalmente este segundo router entrega los datos por la red cableada Ethernet al segundo PC.

A continuación, repetimos todas las medidas, pero utilizando uno de los routers para generar tráfico y enviar las tramas al otro a través de la red inalámbrica (sin que los PCs intervengan). Una vez hemos obtenido las muestras podemos analizar cual a sido el mejor en cada uno de los casos.

Este experimento ha sido realizado con diferentes modelos de routers; usando routers Asus configurados con el modo 802.11g, y utilizando Foneras también en modo 802.11g.

El resultado de los experimentos que realizaremos a continuación le hemos comparado con el máximo teórico esperado [10].

4.1.1 Asus 802.11g vs. PC

En el caso de los ASUS, los dispositivos utilizados para este experimento son CMP110 y CMP111, los cuales han sido configurados para emitir en el canal 9 de 802.11g (radiando a 2472 MHz). La potencia queda fijada siempre a 16 dBm.

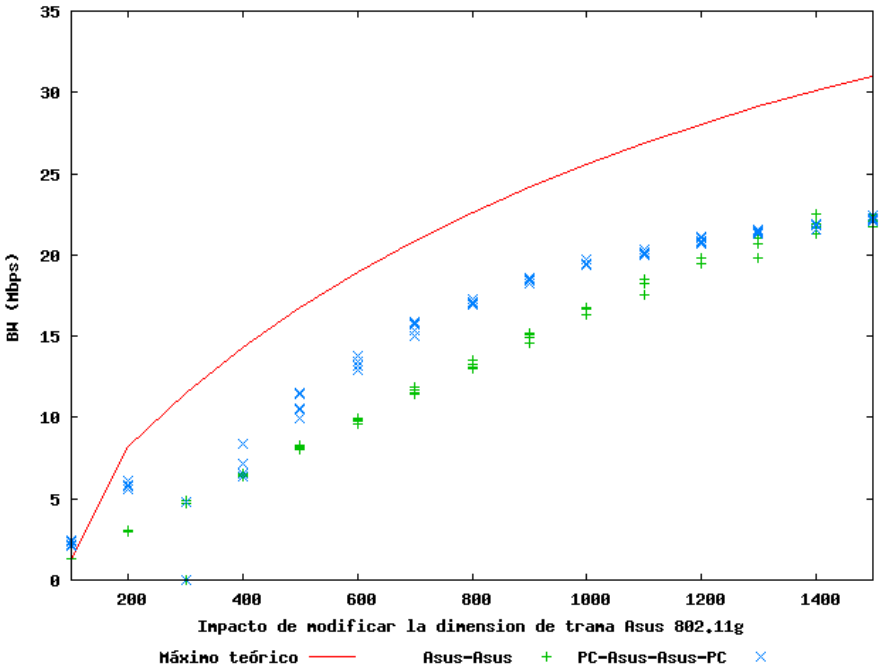


Figura 4.1: Impacto de generar tráfico desde PC o desde los dispositivos Asus en modo 802.11g

En la figura 4.1 podemos observar el resultado de este experimento, en el que vemos que tiene un comportamiento creciente, es decir, según aumentamos el tamaño de trama se incrementa el ancho de banda.

Vemos como para este caso, no importa el elemento que genere el tráfico, ya que para ambos casos tenemos unos valores muy semejantes, de hecho para el valor límite del ancho de trama obtenemos el mismo rendimiento. Creemos que esto es debido a que el interfaz radio esta actuando como cuello

de botella, lo que evita que el comportamiento de la prueba para el caso de generar tráfico con PCs y dispositivos diverja en cuanto al resultado.

En cambio en otro estudio previo [5], si se observan diferencias significativas dependiendo de quien genere el tráfico, para el caso de este mismo modelo de router utilizando el modo 802.11a y también para los Linksys.

4.1.2 Fonera 802.11g vs. PC

Por último realizamos la misma prueba que en la sección anterior, pero en este caso con foneras. Los dispositivos elegidos para tal experimento han sido CMP204 y CMP208, los configuramos de tal manera que el enlace de comunicación inalámbrica tenga una potencia de transmisión de 16 dBm y se lleve a cabo en el modo 802.11g, que es el soportado por las foneras.

Repetimos el experimento 5 veces, variando el tamaño de la trama, y obtenemos el resultado de la figura 4.2.

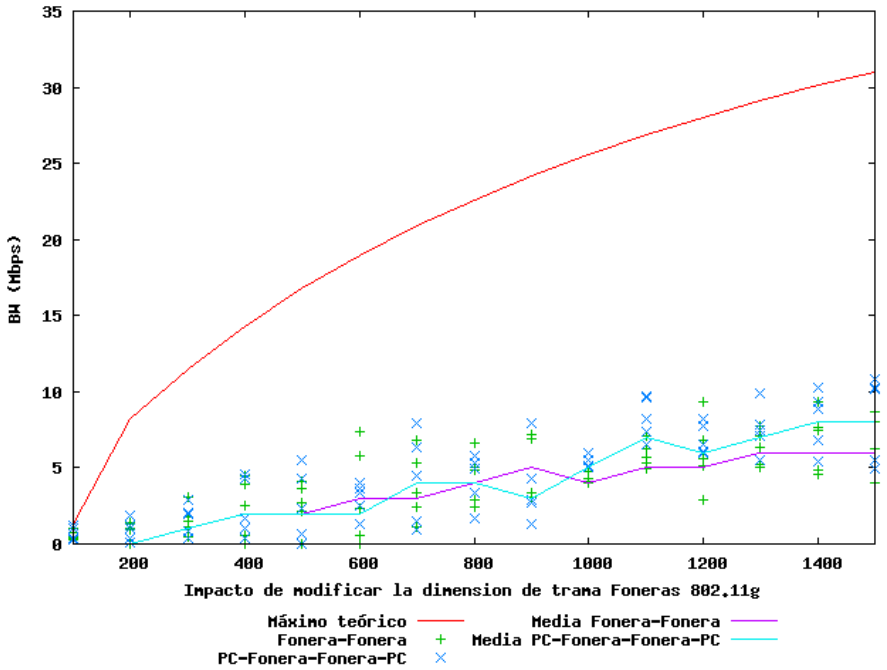


Figura 4.2: Impacto de generar tráfico desde PC o desde las foneras, 802.11g

Podemos ver que el resultado obtenido no es nada bueno en cuanto al rendimiento. Además de las muestras hemos representado en media el resultado

para cada uno de los tamaños de trama. Con esto podemos apreciar mejor que tenemos un ancho de banda muy bajo en cualquier caso.

Aparte de obtener un rendimiento bajo, no se observa un modelo claro en el comportamiento del resultado de esta prueba en cuanto a las mediciones obtenidas para cada una de las cinco repeticiones realizadas. Lo único que se puede ver con ligera claridad es lo que ya hemos visto en los experimentos con otros modelos de routers, y es que subiendo el tamaño de trama, conseguimos obtener un ancho de banda algo mejor.

Ni siquiera generando tráfico desde los PCs se obtiene un rendimiento medianamente bueno, por lo que pensamos que el procesador de este modelo de router no es capaz de procesar tal cantidad de tráfico. Si comparamos la fonera con los otros dos routers estudiados, vemos que esta sale perdiendo en cuanto a características técnicas y por tanto esto se ve afectado a la hora de manejar el tráfico.

Por este motivo, por los problemas comentados anteriormente en el apartado 3.1.3 y por las incompatibilidades encontradas entre distintos firmwares y las foneras comentados en el Anexo B.2.1, se decidió prescindir de este modelo de foneras para la realización de posteriores experimentos en nuestra red de pruebas, además de quedar desaconsejada su instalación para construir redes mesh.

Además de descartar las foneras, hay que decir que nos centraremos únicamente en realizar las pruebas para el modelo de router Asus WL-500g en el modo 802.11g, ya que el modo 802.11a y el router Linksys WRT54GL se estudiaron en trabajos previos a este proyecto [5], [7], en los que se realizaron pruebas similares a las que realizaremos a continuación.

Como conclusión a este experimento, decidimos utilizar de aquí en adelante los PCs para generar tráfico y liberar de esta tarea a los equipos.

El uso de PCs además nos permite disponer de un mayor catálogo de herramientas para generar tráfico, y además utilizándolos conseguimos simular mejor un escenario real de dispositivos accediendo a través de una red mesh. Otra justificación del motivo de usar PCs la encontramos si observamos la gráfica 4.1, vemos que para algunos valores de trama es mejor el uso de PCs y no de dispositivos.

4.2 Estudio del efecto aislante del subsuelo

La red mesh se encuentra desplegada bajo el suelo de un laboratorio en el edificio Torres Quevedo de la Universidad Carlos III de Madrid. Este falso suelo se compone de losetas que son fácilmente extraíbles mediante una ventosa manual, las losetas están fabricadas con madera y además tienen dos pequeñas láminas metálicas cubriendo toda su superficie por ambas caras.

Figura 4.3: Detalle de loseta del falso suelo

Gracias a esta pequeña lámina metálica (ver figura 4.3) tenemos la red en un entorno aislado, y presumiblemente las interferencias externas afectan en menor medida, en especial otras redes 802.11 que nos pudieran molestar para las pruebas de rendimiento.

Para verificar cuanto de efectivo es el aislamiento que nos ofrece el falso suelo vamos a realizar el siguiente experimento con los dispositivos Asus en modo 802.11g:

Tomaremos dos router Asus (CMP111 y CMP112), y los configuraremos con el mismo ESSID, el mismo canal de comunicaciones, y la misma potencia, de esta manera quedarán emparejados. Después con *iperf*, generaremos tráfico UDP, durante intervalos de 30 segundos, a una tasa de envío de 35Mbps. Elegimos esta tasa de envío ya que la tasa máxima en redes de este tipo nunca superará 35Mbps [10]. Repetiremos esta prueba 5 veces en 3 escenarios distintos:

- Escenario A:

  Como vemos en la figura 4.4 pondremos ambos routers bajo el suelo, con este montaje los dos routers y el enlace de comunicación inalámbrico que forman entre ellos quedan aislados de interferencias externas.

Figura 4.4: Ambos routers bajo el suelo

- Escenario B:

  Configuraremos este escenario de la figura 4.5 poniendo un router encima del suelo, y el otro queda debajo.

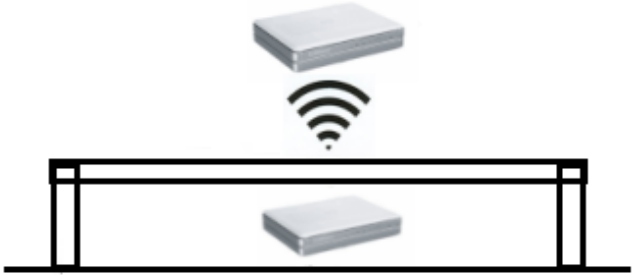


Figura 4.5: Un router encima del suelo, y otro debajo

- Escenario C:

  En este escenario de la figura 4.6 los dos routers se encuentran en el exterior del subsuelo, afectados por las interferencias del entorno, como por ejemplo otras redes inalámbricas 802.11g.

Tras realizar las pruebas comentadas anteriormente, conseguimos los resultado que mostramos en la gráfica 4.7.

En un escenario de comunicaciones común, tendremos el montaje de la figura 4.6, donde los routers se encuentren en un espacio abierto a todo tipo

Figura 4.6: Ambos routers encima del suelo

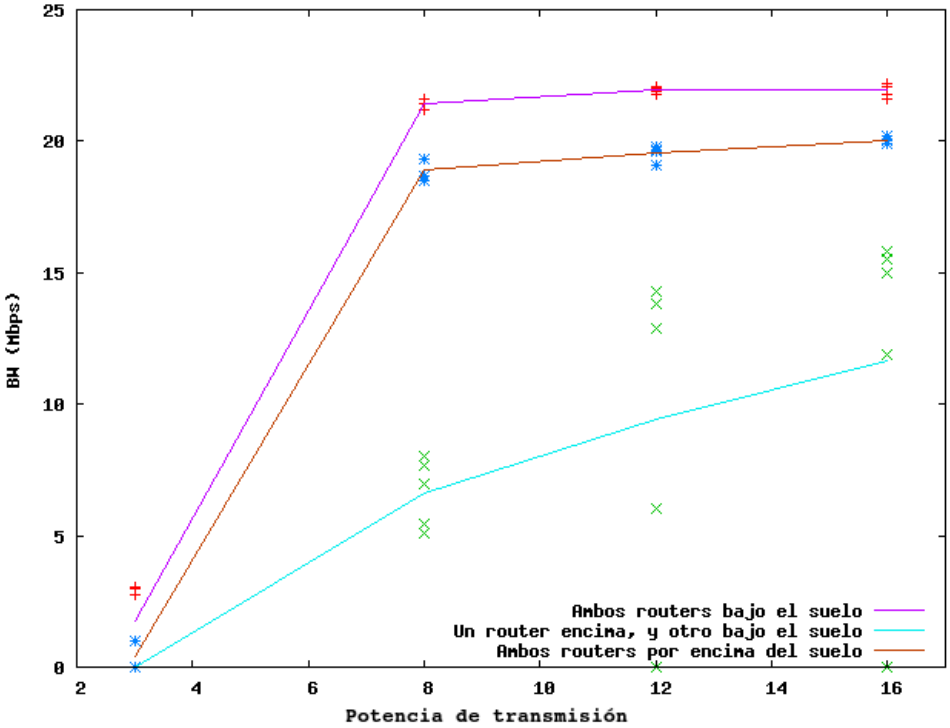


Figura 4.7: Efecto del aislamiento eléctrico del subsuelo

de interferencias, vemos en los resultados que este montaje nos da un rendimiento medio llegando hasta los 20Mbps con la potencia de transmisión a 16dBm.

Como hemos comentado al principio, buscamos mejorar el rendimiento de la red aislándola de cualquier perturbación radioeléctrica externa, por ello introducimos la red de pruebas en un espacio en el que existan interferencias en la menor medida posible. Vemos este caso en el montaje del escenario A, donde según los resultados obtenidos aumentamos el rendimiento de la red en un 10 % respecto al escenario C.

Llegados a este punto vemos que el efecto aislante del suelo, nos beneficia en cuanto al rendimiento, para comprobar si realmente tenemos un aislamiento radioeléctrico vamos a configurar el escenario de la figura 4.5, donde se observa claramente que obtenemos el peor de los rendimientos.

Apenas conseguimos un ancho de banda de 16Mbps en el mejor de los casos, incluso con una potencia de 16dBm. Esto prueba que el efecto aislante del falso suelo es bastante importante ya que la propia comunicación entre los dos routers de la prueba se ve afectada al tenerlos separados por el suelo, llegando a menguar el rendimiento en un 47 %.

4.3 Impacto de la hora del día

A la hora de caracterizar la red de pruebas, es interesante comprobar si existen más interferencias a ciertas horas del día que a otras. Interferencias provenientes de otras redes inalámbricas cercanas a la nuestra, o incluso cualquier dispositivo que se encuentre radiando en nuestra banda de frecuencias.

Con el objetivo de comprobar si las prestaciones dependen de la hora del día en que utilicemos la red, vamos a realizar un experimento, el cual durará 24 horas durante un día normal de trabajo entre semana. El resultado de este experimento será crucial para realizar los experimentos posteriores en un entorno ideal y con las menores interferencias posibles.

Comenzaremos la prueba, creando un enlace unidireccional, entre dos dispositivos Asus configurados en el modo 802.11g. Para comprobar el ancho de banda utilizaremos la herramienta *iperf* con la que generaremos e inyectaremos tráfico a través de la red desde uno de los PCs hasta el otro. Enrutaremos el tráfico de tal manera que desde el PC origen envíe los datos por Ethernet a uno de los router Asus, este enviará los datos a través de su interfaz inalámbrico al otro router Asus, el cual a su vez enrutará con el otro PC de destino por medio de la red cableada. El tráfico generado consiste en un flujo de datos UDP a 35 Mbps, usando tramas de 1500 bytes durante un intervalo de 30 segundos.

Además hemos utilizado el nodo CMP110 como un nodo sonda, cuya función será detectar el número de tramas que hay en el entorno durante la

realización de la prueba. Hemos configurado el nodo en modo monitor, y con la herramienta *tcpdump* capturaremos el número de tramas de otras redes inalámbricas, para ello le hemos aplicado un filtro que discrimina las tramas de nuestros equipos de pruebas. Gracias a esto observamos el tráfico proveniente de otras redes cercanas.

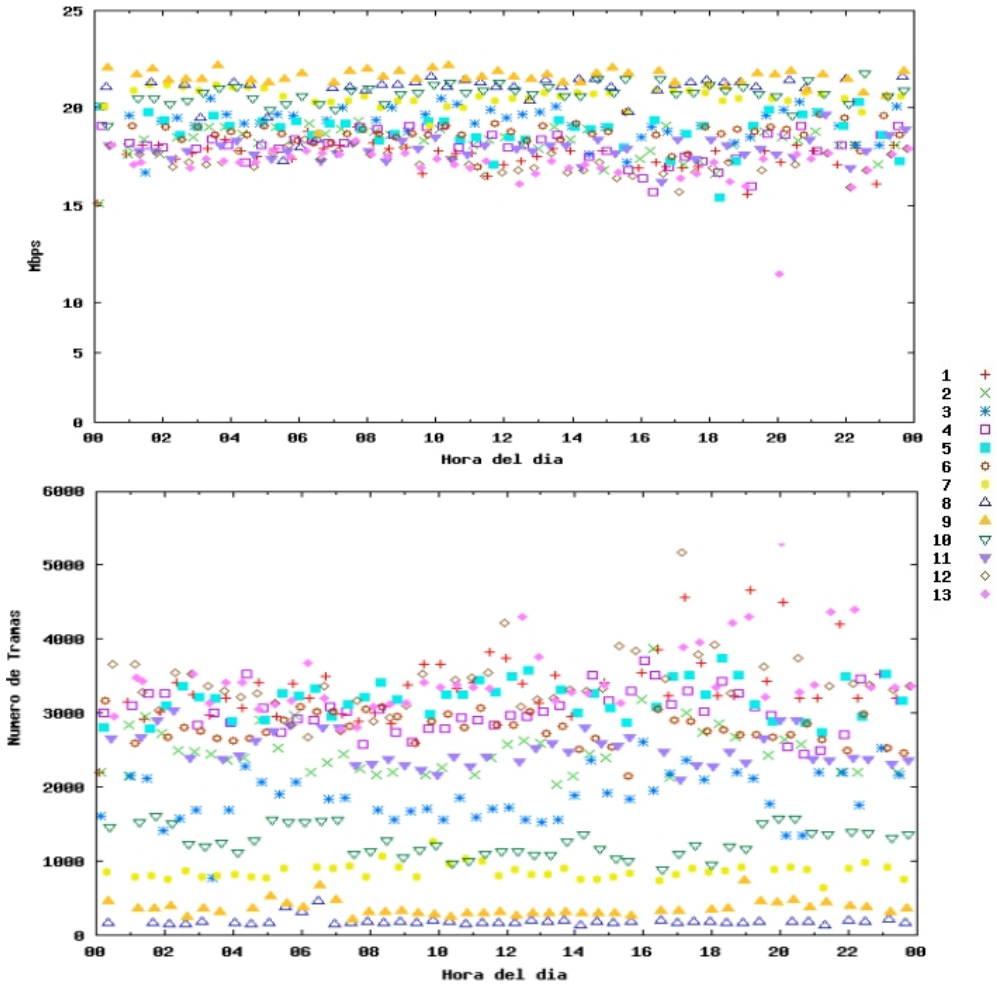


Figura 4.8: Impacto de la hora del día en routers Asus 802.11g

La figura 4.8 representa para cada uno de los canales del modo 802.11g, en la parte superior los distintos rendimientos obtenidos durante la medición a lo largo de las 24 horas del día. En la parte inferior se representa el número total de tramas capturadas por nuestro nodo sonda y posteriormente filtradas discriminando las de nuestra propia red.

Podemos ver como en las horas centrales del día hay un ligero incremento en cuanto al número de paquetes que circulan en nuestro entorno de pruebas,

lo que apenas afecta al rendimiento de la red.

Se puede comprobar como la red tiene un comportamiento bastante lineal, apenas hay dispersión de los datos que se quedan agrupados en torno a 20 Mbps, parece que el efecto de variar el canal es más considerable que el de utilizar la red a ciertas horas del día.

Aunque para algunos canales como por ejemplo el 13 (2472 MHz.) o el 11 (2462 MHz.) sí parece que afecte el hecho de ser usados a ciertas horas del día, sobre todo a partir de las 9 de la mañana. Esto es debido a que a partir de esta hora hay más tráfico de datos inalámbricos debido al horario de trabajo de la universidad, podemos ver como a partir de las 20 horas hay un ligero incremento de la calidad en los canales comentados.

Con el objetivo de esclarecer un poco más el resultado de la prueba hemos calculado los siguientes parámetros estadísticos. En la tabla 4.1 podemos ver cual es el rendimiento medio obtenido de todas las muestras para todos los canales utilizados en la prueba, así como la media del número de tramas que hubo en la red durante el tiempo que duró la misma.

Hemos calculado también la varianza de todas las muestras, con este dato nos hacemos una idea de la desviación que han tenido respecto a su media, nos sirve para deducir si el resultado ha sido uniforme en cada repetición o por el contrario hemos tomado muestras muy distintas unas de otras. En nuestro caso tenemos un valor de 0.53 Mbps por lo que las muestras están bastante uniformemente obtenidas.

Podemos ver además el índice de correlación, el hecho de que sea negativo, indica una dependencia total entre las dos variables llamada relación inversa: cuando una de ellas aumenta, la otra disminuye en idéntica proporción. Esto corrobora lo que ya sabíamos, que al tener más tramas en el entorno de pruebas nos introduce ruido en nuestra red y por tanto empeorará el rendimiento.

Durante el experimento se ha ido cambiando la frecuencia a la que realizamos la prueba. Hemos tomado muestras en cada uno de los canales del modo 802.11g, comenzando en el canal 1 (2412 MHz.) y acabando en el canal 13 (2472 MHz.). Este efecto, como comentábamos anteriormente, parece que es más determinante que el hecho de medir la calidades de la red a unas

Cálculos estadísticos de la figura	
Media BW:	18.9 Mbps
Varianza BW:	0.53 Mbps
Media del nº Tramas:	2910
Correlación:	-0.73

Tabla 4.1: Estadísticos de la prueba de 24 horas

horas u otras. Para ver mejor el impacto de utilización de ciertos canales, hemos generado la gráfica de la figura 4.9.

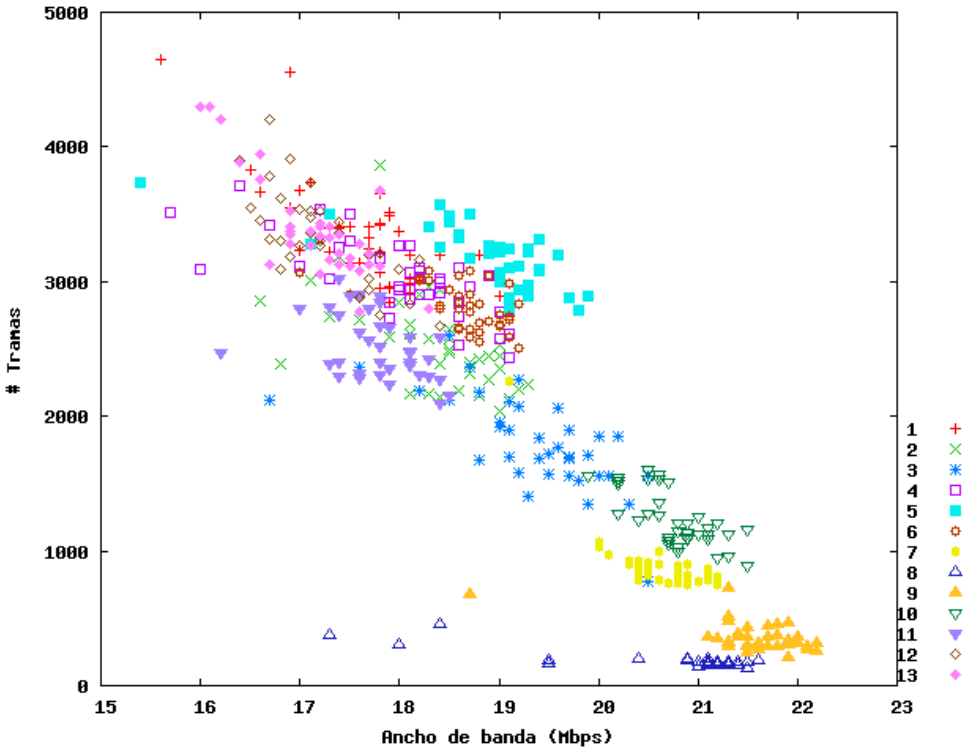


Figura 4.9: Rendimiento en función de los canales en Asus 802.11g

En esta figura se ve claramente como el número de tramas en los canales 8 y 9 es casi nulo, por lo que se tiene un rendimiento bastante alto. En cambio para los canales 11, 12 y 13 se observa que el número de tramas en nuestro entorno de pruebas era bastante alto, cosa que afectó al rendimiento de la red.

4.4 Impacto de la potencia de transmisión

Una de las principales ventajas de nuestra red de pruebas, es que se pueden simular una gran variedad de escenarios multisalto. En esta sección se pretende ver, en qué medida afecta a la conectividad de la red, el hecho de variar la potencia de transmisión de sus elementos.

De los posibles enlaces que se pueden formar en la red de pruebas, nos interesa averiguar cuantos de ellos funcionan con un cierto rendimiento si jugamos con la potencia. Para ello realizamos el siguiente experimento:

Primero, configuraremos cada uno de los dispositivos Asus, en el modo 802.11g, utilizando la misma potencia de transmisión en cada uno de ellos. Con esto tendremos *N* nodos y *N* x *(N-1)* enlaces posibles entre ellos. Para cada uno de los enlaces vamos a medir el ancho de banda con tráfico UDP, durante 30 segundos y usando el mismo canal de comunicaciones. Con esto obtendremos el rendimiento de cada uno de los 14 nodos respecto a los otros 13, teniendo en cuenta que cada enlace será activado independientemente de los demás, cada uno a un tiempo, con el objetivo de que no se molesten entre sí.

Con este experimento realizado en 14 nodos tenemos un total de 182 enlaces unidireccionales, repetimos la prueba 5 veces y promediamos el resultado entre las distintas repeticiones. Después de todo esto, ordenamos de mayor a menor los 182 resultados y preparemos los datos para ser presentados en una gráfica.

Esta prueba se ha realizado para distintas potencias de transmisión (4,7,9,11,13 y 16 dBm), y el resultado final se representa en la gráfica de la figura 4.10.

Se observa como para una potencia baja (4dBm), apenas hay algo más de 40 enlaces (de los 182), que obtengamos resultados distintos de cero, en cambio para potencias altas (13,16 dBm) se ve como la mayoría de los enlaces se encuentran en torno a un rendimiento de 20Mbps.

Podemos deducir con la figura, que para este modo 802.11g, usando varios niveles de potencia de transmisión, sería posible modificar la conectividad de los diferentes nodos de nuestra red de pruebas, ya que hay una gran variedad entre las distintas potencias que utilizamos.

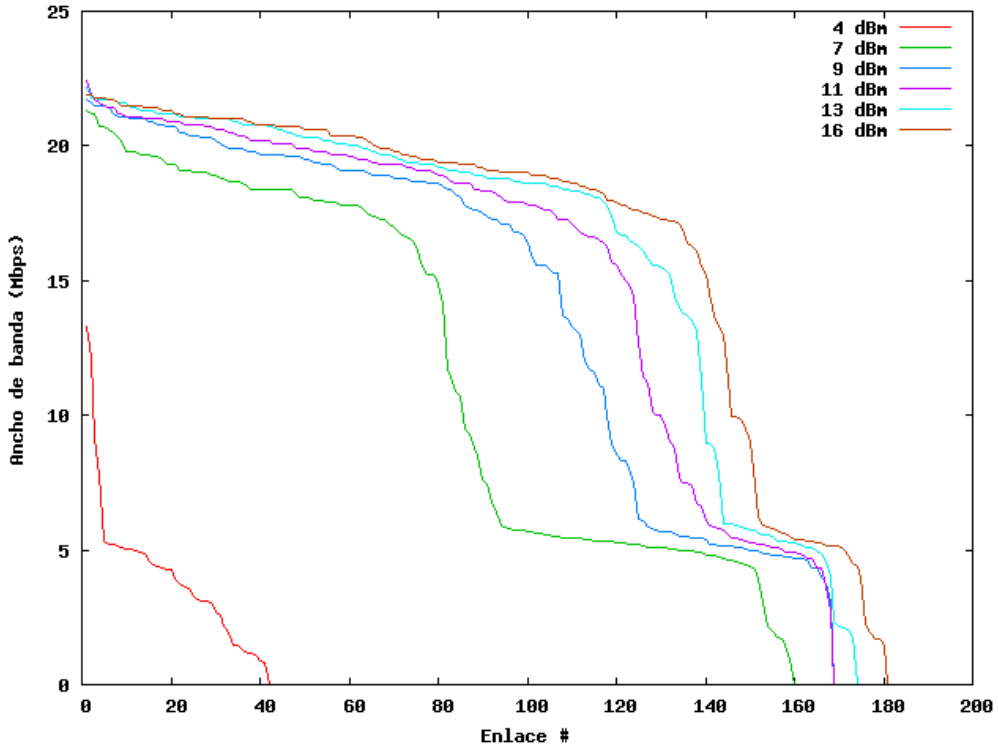


Figura 4.10: Impacto de la potencia de transmisión en la red de pruebas con Asus 802.11g

La conclusión principal a la que llegamos es que, en nuestra red de pruebas podemos usar los niveles de transmisión, para configurar distintos enlaces, resultando con ello una gran diversidad de topologías de redes multisalto.

4.5 Impacto de la interferencia en canales 802.11b/g adyacentes

Hasta ahora hemos hablado de pruebas realizadas entre dos únicos equipos, en este apartado se van a realizar pruebas con dos enlaces al mismo tiempo, con lo que tendremos que tener en cuenta la interferencia que cada enlace generará respecto al otro.

Como podemos ver en la figura 4.11 hemos considerado dos posibles escenarios para la realización de esta prueba:

- Escenario A: Nodos interferentes lejanos. El enlace que pretender ser

interferente se encuentran a una distancia relativa mayor que los nodos del mismo enlace.

- Escenario B: Nodos interferentes cercanos. En este caso la distancia relativa entre los nodos de un mismo enlace es mucho mayor, al contrario que los nodos interferentes, que están bastante cerca.

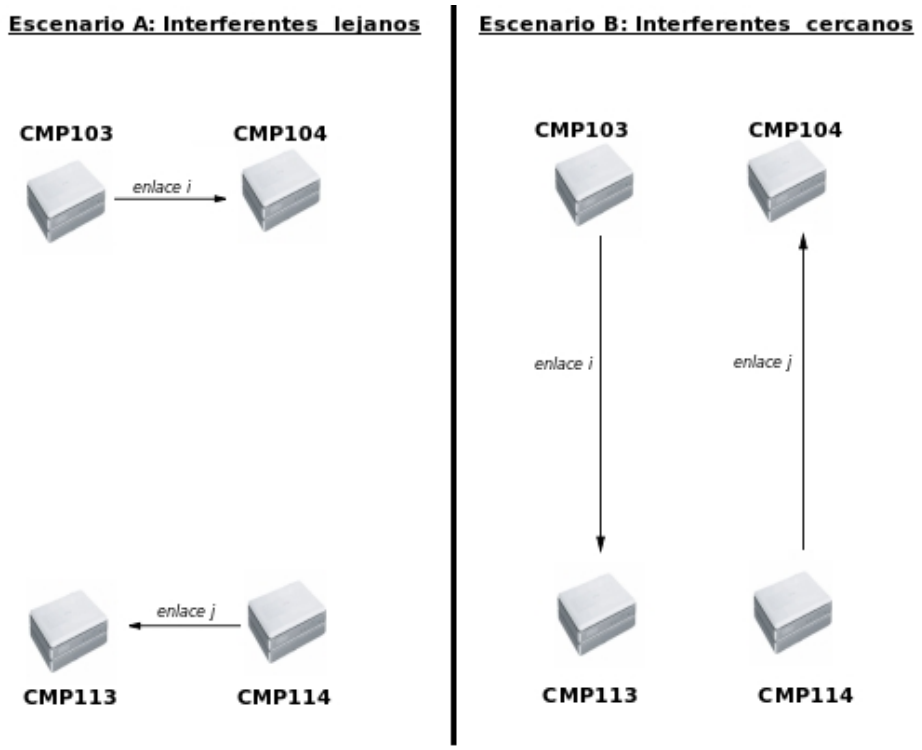


Figura 4.11: Escenarios de la prueba

A la hora de situar los elementos para esta prueba, conviene tener en cuenta el efecto de campo cercano de las antenas de los dispositivos inalámbricos de los routers, este efecto es indeseable ya que nos produciría inducciones electromagnéticas entre las antenas, lo que aumentaría el nivel de ruido durante las mediciones y por tanto bajaría la calidad del enlace.

El límite de este campo cercano viene dado por la siguiente formula[11]:

$$(l = \frac{2D^2}{\lambda})$$

donde D es el diámetro de la antena y λ es la longitud de onda de la frecuencia de transmisión utilizada.

Mediante esta fórmula evaluamos si las antenas están lo suficientemente separadas para no tener los efectos de campo cercano que siempre queremos evitar, por lo que las distancias absolutas entre ellas deberán ser mucho mayor que el límite”l“.

4.5.1 Escenario A: interferentes lejanos

Antes de comenzar la prueba, conviene recordar cual es la distancia teórica mínima entre canales en el modo 802.11b/g para no encontrarse solapados. Como podemos ver en la figura 2.10 el ancho de banda de la señal inalámbrica es de 22Mhz, y los canales están separados 5Mhz en la banda de frecuencias, por tanto la hemos de dejar como mínimo 4 canales entre medias, esta separación es necesaria para suponer los canales libres de interferencias. De aquí en adelante denominaremos esta distancia entre canales como ”d“.

Configuraremos nuestros dispositivos en modo 802.11g y con tres distancias distintas, de la siguiente manera:

- 1) Ambos enlaces i y j se configuran en el canal 13, por tanto la distancia será igual a cero.
- 2) El enlace i se configurará en el canal 13 y el enlace j en el canal 8, por lo que d=5.
- 3) Ambos enlaces quedan separados una distancia d=10, ya que serán configurados en los canales 13 y 3 respectivamente.

  Decidimos utilizar estos 3 canales por la condición de que tienen que estar separados la distancia que hemos diseñado para la prueba y además como vimos en la prueba anterior (4.3) se encuentran bastante libres de interferencias de otras redes del laboratorio.

Además de todo lo anterior, para probar la influencia de la potencia de transmisión, la modificaremos desde 3dBm hasta 15dBm en pasos de 2dBm, y tomaremos la medida de dos tasas de rendimiento de la red; una de ellas será el ancho de banda medido en el enlace i, con el enlace j apagado,y viceversa. La otra medida será el ancho de banda del enlace i y del enlace j ambos radiando a la vez.

El resultado de esta prueba le podemos observar en la figura 4.12, donde vemos sumados los anchos de banda cuando estamos transmitiendo con los enlaces cada uno a un tiempo $(Ri_{solo} + Rj_{solo})$ y también en la figura 4.13 con ambos enlaces a la vez $(Ri_{juntos} + Rj_{juntos})$.

Para realizar estas gráficas se ha repetido la prueba 5 veces y posteriormente se han promediado cada una de las 5 muestras, obteniendo así la línea que representa la media de los datos para cada potencia.

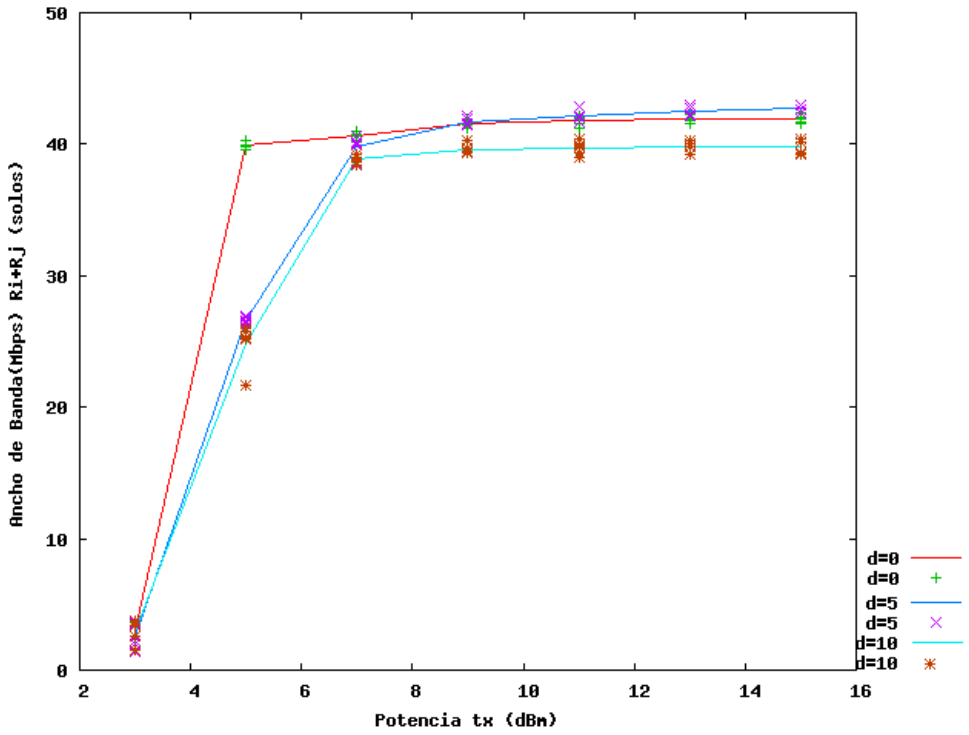


Figura 4.12: Enlaces radiando por separado con Asus 802.11g en el escenario A

Significativamente se observa como los routers tienen problemas si se encuentran configurados con una potencia de transmisión de 3dBm, pero en cuanto ampliamos la potencia a 5 y a 7 dBm el rendimiento se ve rápidamente beneficiado, llegando a cuadruplicar el nivel del ancho de banda.

Conviene hacer notar que el efecto de configurar los canales separados una distancia d, no se tiene en consideración en la gráfica 4.12, ya que al encontrarse radiando cada enlace a un tiempo, no se molestarían uno al otro, y por tanto el entorno estaría libre de otro canal que pudiera interferir la comu-

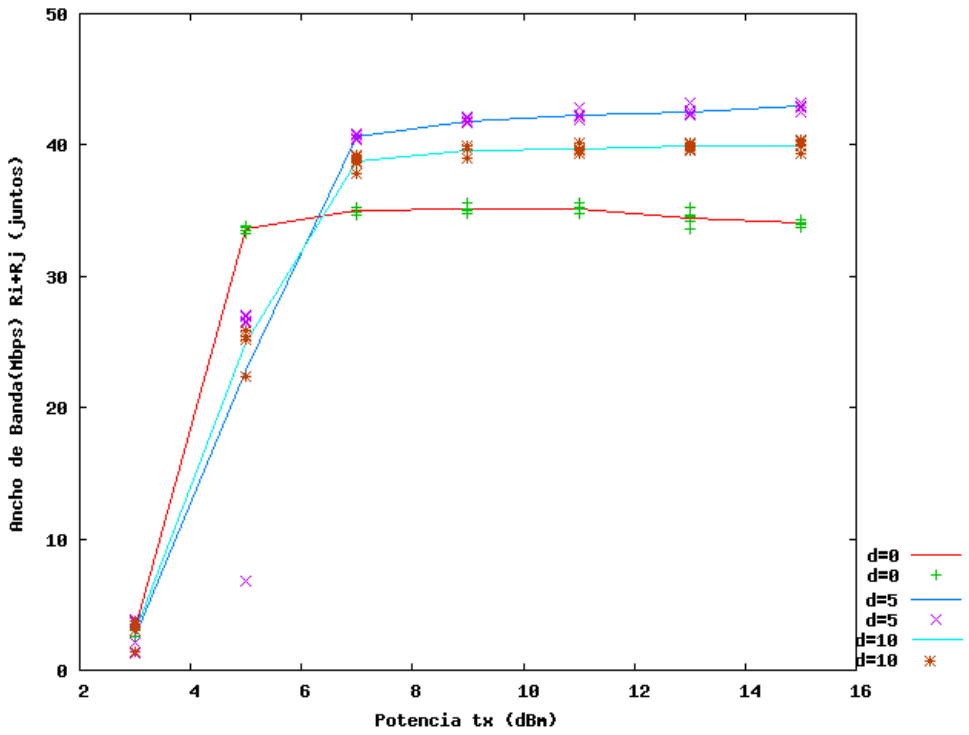


Figura 4.13: Enlaces radiando juntos con Asus 802.11g en el escenario A

nicación.

De la gráfica 4.13 podemos sacar la conclusión de que cuando se encuentran radiando ambos enlaces a la vez, apenas se obtiene diferencia en cuanto al rendimiento, a no ser que los dos se encuentren en el mismo canal, es decir, si los configuramos con distancia de separación d=0 . En este caso se ve como la calidad del enlace se ve afectada en un 25 % por el ruido que introduce el otro enlace.

Hemos colocado físicamente los Asus a una distancia lo suficientemente grande entre los enlaces interferentes y pensamos que esto hace que no se molesten entre ellos y el enlace únicamente se vea degradado en un 25 %.

Al separar los canales una distancia d=5 (no solapados), vemos como apenas interfiere un canal en otro, ya que el rendimiento mejora notablemente para este caso.

Ya se ha estudiado anteriormente en otras publicaciones [7] que esto no ocurre así con otras marcas comerciales como Linksys WRT54GL, en este modelo de router incluso con distancias d=10 se ve el efecto de como se sola-

pan los canales y donde si se obtiene un rendimiento del 50 % cuando ambos enlaces se encuentran funcionando a la vez.

Eficiencia de la separación de canales

Para comprobar más a fondo el efecto de solapamiento entre canales con los routers Asus WL-500, configurados en el modo 802.11g, realizaremos otro experimento con el que comprobaremos la calidad de los filtros cuando estamos radiando con dos enlaces separados distancias d=0,3,5,10. Lo haremos para enlaces interferentes lejanos y ambos a la vez, de esta manera podremos comparar los resultados con el experimento anterior.

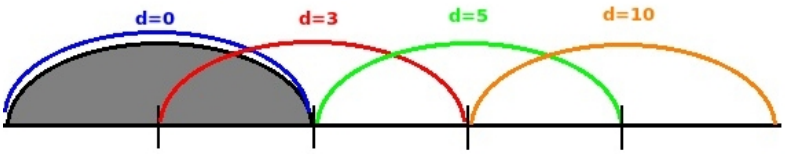


Figura 4.14: Separación entre canales d=0,3,5,10

En el esquema de la figura 4.14, se repesentan los distintos casos de la prueba:

- En el caso de d=0, tendremos un solapamiento total por lo que es de esperar que el rendimiento empeore bastante respecto a la anterior prueba cuando los enlaces radiaban cada uno a un tiempo.
- Para d=3 tenemos también un solapamiento aunque algo menor que el anterior.
- Con d=5, si el filtro es lo suficientemente bueno, se verá que el rendimiento mejora en el caso en que ambos enlaces se encuentren radiando a la par.
- Por último configuraremos una distancia ”d”lo suficientemente lejana como para asegurarnos que ambos enlaces no se solaparán en frecuencias, esta será d=10.

Para tratar de representar mejor, y sea más fácilmente entendible, utilizaremos la siguiente expresión donde calcularemos el parámetro η.

$$(\eta = \frac{Ri_{juntos} + Rj_{juntos}}{Ri_{solo} + Rj_{solo}})$$

Definiremos η como la eficiencia resultante al utilizar una separación de canal concreta. De esta forma, η tomará valores desde 0, donde tendremos un efecto muy fuerte de la interferencia, hasta 1 donde la interferencia será prácticamente nula.

Realizamos la prueba en las condiciones establecidas anteriormente y representamos los valores de η en la gráfica de la figura 4.15.

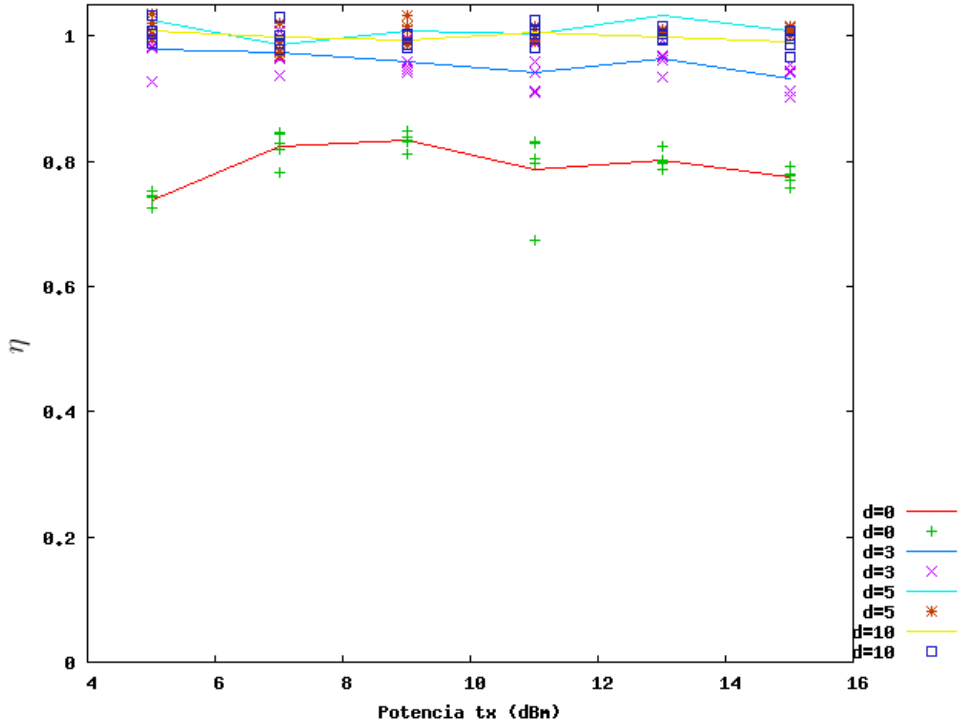


Figura 4.15: Interferentes lejanos 802.11g: (η) eficiencia de la separación de canales

Se puede ver como para unas frecuencias separadas una distancia d=5 o mayor, el efecto de la interferencia apenas afecta a las condiciones de la prueba ya que obtenemos valores muy cercanos a η=1, según disminuimos la distancia entre canales vemos como η se aleja de 1, si llevamos al límite este experimento, configurando los canales solapados totalmente (d=0) se ve

como las interferencias hacen mella en el rendimiento del enlace, y nos da unos valores de η entorno a 0.75, lo que hace que podamos afianzar nuestra anterior afirmación, donde decíamos que la calidad del enlace, en este caso, se ve afectada un 25 %.

El caso de una separación d=3, es aquel en el que hay cierta interferencia entre canales ya que como observamos en a la figura 4.14 estos se encuentran parcialmente solapados. Vemos según los resultados que a pesar de esta situación, la eficiencia de separación de canales (η) sigue siendo bastante buena, en torno a 0.9. Pensamos que esto se debe al efecto distancia física que existe entre las antenas de los emisores, existiendo una reutilización espacial de frecuencias.

El efecto de reutilización espacial de frecuencias parece que es más determinante para las frecuencias que utilizan este modo 802.11g, que para las del 802.11a. En el modo 802.11a, como ya se estudió en otro trabajo previo [5], para unos enlaces con interferentes lejanos radiando a una potencia de 15 dBm el parámetro η estaba en torno a 0.5, y en nuestro caso este valor es de 0.75.

Creemos que esta diferencia es debida principalmente al hecho de que con las frecuencias más altas (del rango de 5Ghz, que usa el modo 802.11a) se consigue llegar más lejos (aunque con menos potencia, cosa que aquí no influye si usamos potencias medias/altas ya que estamos en un entorno reducido) que con las frecuencias del entorno de los 2.4 MHz y por tanto se interfieren entre sí a pesar de esta separados.

4.5.2 Escenario B: interferentes cercanos

Los experimentos del apartado anterior, han sido realizados configurando los nodos interferentes como lejanos, (el nodo interferente más lejano que el nodo destino). Ahora configuraremos el escenario que se indica en la figura 4.11, en el modo interferentes cercanos, esto es, poniendo el nodo interferente más cerca que el propio nodo de destino.

En las figuras 4.16 y 4.17, hemos representado los resultados obtenidos de los experimentos realizados en el apartado anterior, pero para nodos inter-

ferentes cercanos.

Al ver las gráficas de lo primero que nos damos cuenta es que la potencia juega un papel más importante que en el experimento anterior. Esto se debe a la mayor distancia física entre elementos, el configurar el enlace con una potencia alta nos hace tener un rendimiento mucho mejor que si lo configuramos a una potencia media.

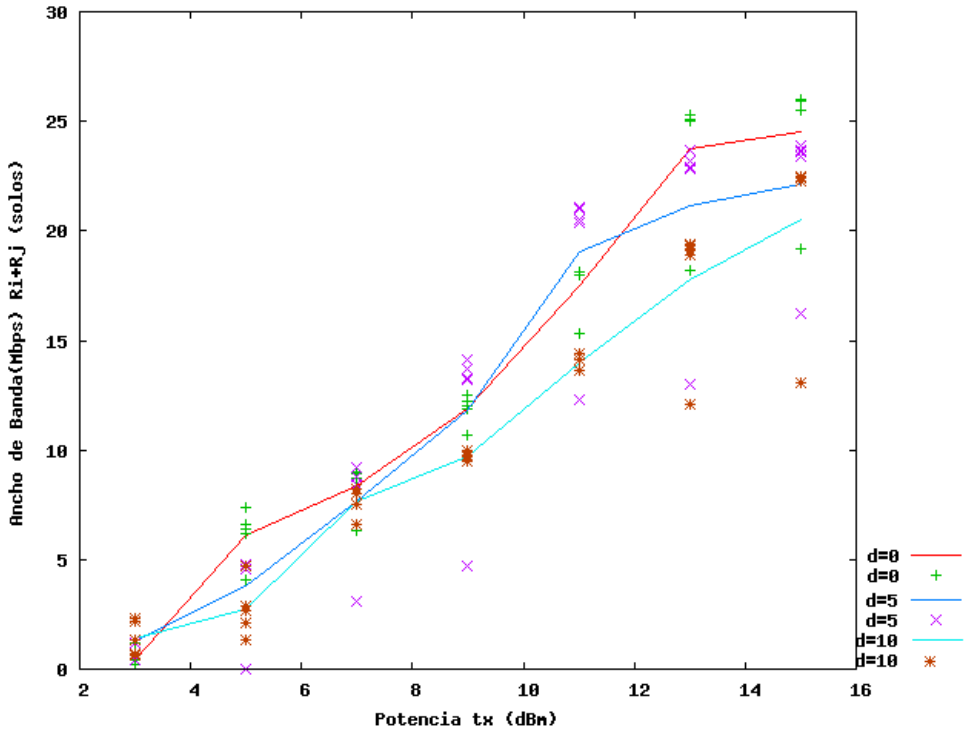


Figura 4.16: Enlaces radiando por separado con Asus 802.11g en el escenario B

Igual que en el experimento diseñado en el apartado anterior, la figura 4.16 es el resultado de poner en marcha ambos enlaces por separado $(Ri_{solo} + Rj_{solo})$.

En dicha figura se observa como para potencia bajas, en torno a 3dBm, apenas llegamos a 2Mbps. Además según ampliamos la potencia, vamos creciendo también en rendimiento, se observa en la gráfica una dependencia muy lineal entre la potencia y el rendimiento. Finalmente llegamos a configurar la potencia máxima para la prueba que eran 15dBm y obtener un rendimiento en torno a 22Mbps.

La diferencia de rendimiento entre distintas distancias ”d“, no se acentúa hasta que no estamos en valores altos de potencia. Ya comentamos anteriormente que, para enlaces comunicándose alternativamente, este efecto no debería evaluarse, y no tener en cuenta la distancia entre los canales.

A pesar de ello se ve una diferencia considerable de rendimiento (4 Mbps) entre las separaciones d=10 y d=0. El que ocurra esto se puede explicar con lo que sucedía en el experimento 4.3, en el que quedó demostrado que el usar un canal de comunicaciones u otro nos hace tener mejor o peor rendimiento. De hecho aquí para d=10, utilizamos el canal 13, que era con el que más bajo rendimiento se obtenía en los resultados de este experimento 4.3.

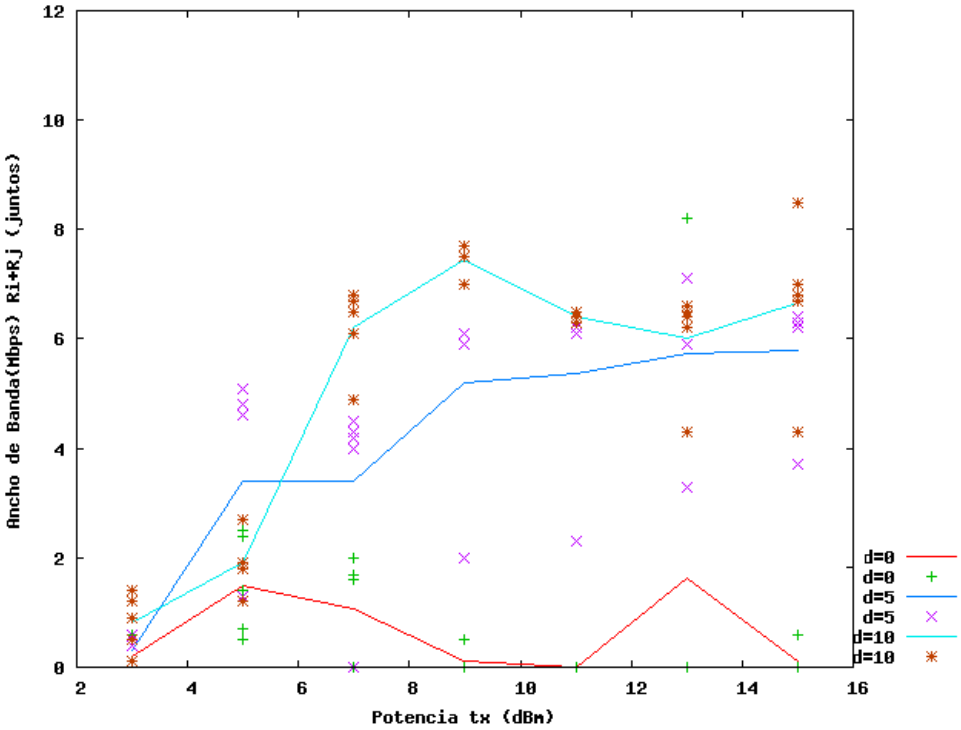


Figura 4.17: Enlaces radiando juntos con Asus 802.11g en el escenario B

En cuanto a la figura 4.17, que representa ambos enlaces funcionando a la vez, $(Ri_{juntos} + Rj_{juntos})$, podemos ver que los resultados se comportan de una manera poco ordenada. Se puede ver que hay una gran dispersión de las muestras tomadas durante la realización de la prueba.

No parece que exista una relación muy clara entre la potencia y el rendimiento, ya que por ejemplo para una potencia intermedia en torno a 9dBm,

es donde tenemos los mejores resultados si ponemos los canales ortogonales (no solapados), y en cambio para potencias mayores bajamos el rendimiento en casi un 25 %.

Vemos que se produce el efecto que comentamos anteriormente, a bajas potencias existe un menor nivel de interferencia y a medida que sube la potencia disminuimos el rendimiento de la prueba. Observamos además que si hay solapamiento de canales, pues el experimento tiene resultados bajos, incluso para una distancia de separación de 10 canales.

En el caso que los canales se encuentra totalmente solapados (d=0), vemos que el rendimiento de la red apenas llega a 1Mbps en el mejor de los casos, es decir, los canales solapados se interfieren uno al otro produciendo casi la anulación de la comunicación entre los elementos de ambos enlaces.

Un efecto que se ve claramente es que el tener una potencia baja (3dBm), hace que a pesar de tener nodos interferentes cercanos los canales no se molesten entre sí, y por tanto, carezca de sentido el separar los canales o no, ya que obtendremos un rendimiento muy parecido en cualquier caso.

Como conclusión podemos decir que, esta manera de configurar la red, sería la menos óptima de todas, ya que tenemos dos circunstancias agravantes, la primera es que tenemos interferentes muy cercanos y la segunda es que los elementos emisores-receptores se encuentran bastante lejanos. Esto se ve reflejado en el rendimiento de la prueba, en cuyo mejor caso apenas llega a 8Mbps.

Eficiencia de la separación de canales

Para ver como afecta el tener un nodo interferente cerca, hemos calculado al igual que para los nodos interferentes lejanos, el valor de (η) en la gráfica 4.18. Recordemos que:

$$(\eta = \frac{Ri_{juntos} + Rj_{juntos}}{Ri_{solo} + Rj_{solo}})$$

η por tanto tenderá a 1 sí tenemos unos niveles de interferencia bajos, y tenderá a 0 si los niveles de interferencia son altos.

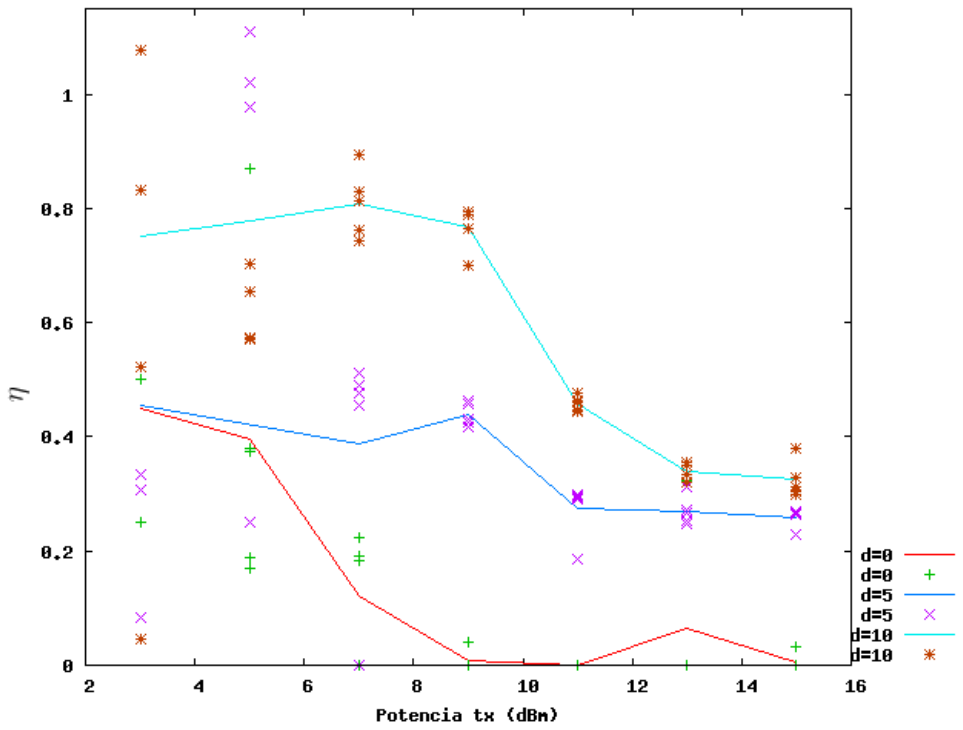


Figura 4.18: Interferentes cercanos 802.11g: (η) eficiencia de la separación de canales

Según los resultados obtenidos observamos una clara relación entre el nivel de potencia y el de interferencia. Podemos decir que nivel de interferencia es aceptable cuando utilizamos potencias bajas, y se va acentuando según subimos esta, hasta llegar al punto de tener una interferencia casi total cuando usamos canales solapados totalmente.

Podemos comprobar el efecto de la interferencia con la gráfica 4.17, vemos que hay una cierta relación inversa entre el rendimiento de esta gráfica y el valor de η. Observamos que cuando el rendimiento aumenta, es precisamente porque la interferencia ha bajado ($\eta \longrightarrow 1$), y ocurre lo contrario si la interferencia aumenta ($\eta \longrightarrow 0$), afecta al rendimiento, el cual baja.

Como conclusión a este experimento, comparando los resultados de esta gráfica 4.18 con la anterior 4.15, podemos decir que el instalar físicamente los nodos de la red correctamente parece que tiene cierta importancia, y no solo el hecho de utilizar unas frecuencias u otras para configurar los canales del enlace, sin importar tanto si estas se solapan o no.

4.6 Medidas en red Multisalto

Finalizados los experimentos anteriores, en los que realizamos pruebas con uno y dos enlaces, damos un paso más en nuestras pruebas y configuramos la red inalámbrica mallada multisalto, utilizaremos para esto los routers Asus WL-500g en el modo 802.11g.

Esta topología de red es bastante interesante por las ventajas que tiene en instalaciones reales. Como vimos en el apartado 4.4, es posible realizar esta configuración gracias al uso de distintas potencias para crear los distintos enlaces, utilizando potencias altas tenemos una gran cantidad de enlaces posibles en la red.

Comenzaremos la prueba configurando las tablas de rutas de los dispositivos y de los PCs para conseguir la topología mostrada en la figura 4.19, que consiste en dar 3 saltos inalámbricos a través de la red. Hay que tener en cuenta que el número de saltos inicial, se reduce a 3 por los problemas de solapamiento de canales adyacentes en el modo 802.11g (estudiado en la sección 4.5), ya que queremos obtener el mejor resultado idealizado a una red multisalto sin interferencias entre sus propios enlaces.

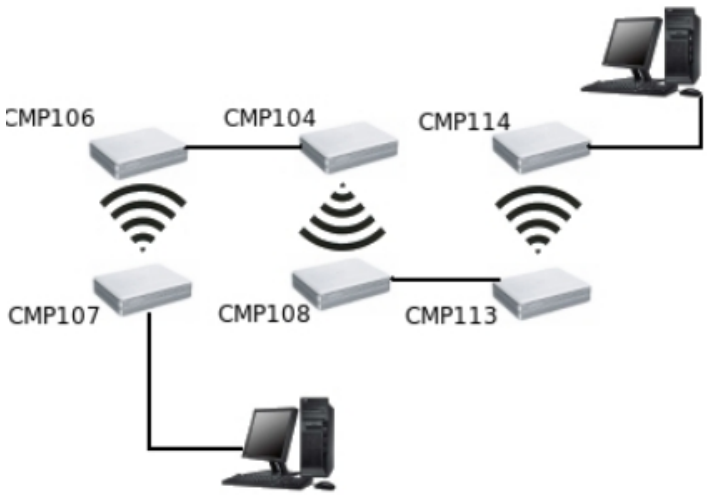


Figura 4.19: Escenario de red multisalto con 3 saltos. Asus 802.11g

Posteriormente buscaremos mediante un algoritmo heurístico, la mejor

combinación posible para llegar secuencialmente hasta los 7 saltos que se muestran en el escenario de la figura 4.20. Este algoritmo buscará la mejor opción de canales y potencias hasta llegar a obtener el mejor resultado, en cuyo caso suponemos que siempre será igual o peor que el obtenido en el paso anterior donde teníamos un caso ideal sin interferencias con 3 saltos.

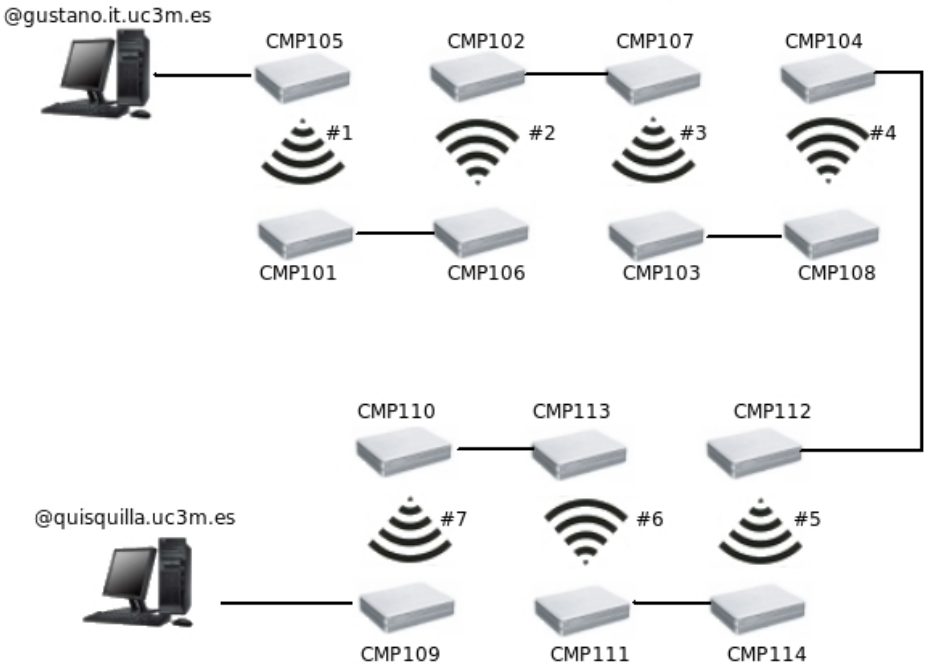


Figura 4.20: Escenario de red multisalto con 7 saltos inalámbricos. Asus 802.11g

Hay que tener en cuenta en ambos casos que el máximo rendimiento que será posible alcanzar será el que obtuvimos con enlaces simples, es decir, unos 23 Mbps. (figura: 4.8).

Comenzamos por tanto realizando el escenario de la figura 4.19, donde configuramos enlaces entre los routers con los valores que a priori parecen más óptimos para realizar la prueba y obtener el máximo rendimiento.

La potencia a 16 dBm es con la que mejor rendimiento se obtuvo en el apartado 4.4 del proyecto. Además usaremos 3 tipos de separaciones entre canales, canales solapados, canales parcialmente solapados (d=3), y por último usaremos unos canales que no se interfieran entre ellos(d=5). Hacemos esto

ya que no tenemos experiencias previas en escenarios con 3 saltos y no sabemos como se comportará la red, así tenemos bajo estudio un mayor número de casos.

Hemos realizado la prueba, como siempre, inyectando tráfico UDP en la red durante 30 segundos.

El resultado que obtenemos de realizar dicha prueba le vemos en la gráfica de la figura 4.21, donde podemos observar como comenzamos con un ancho de banda aceptable si utilizamos canales no solapados (d=5), y de hecho la calidad se mantiene bastante bien (en torno a 19 Mbps) aunque vayamos introduciendo enlaces.

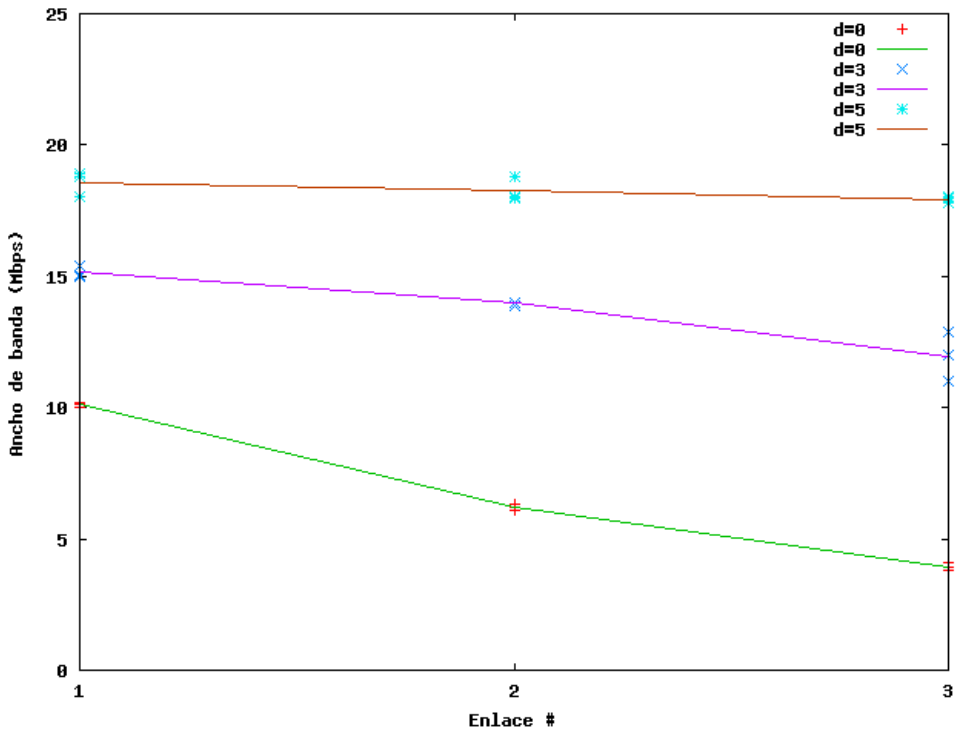


Figura 4.21: Medida del ancho de banda en red multisalto con 3 saltos

Vemos que para unos enlaces con canales parcialmente solapados afecta algo más que en caso d=5 el hecho de dar un salto inalámbrico en la red, en el primer salto el rendimiento del enlace estaba en torno a 15 Mbps y al dar el tercer salto terminamos con 12 Mbps.

Con canales interfiriendose entre si totalmente, comenzamos con una calidad del enlace de 10 Mbps en el primer salto, y llegamos al tercero con

apenas 4 Mbps.

Si comparamos esta prueba con la que realizamos anteriormente (4.5) vemos que existe un ligero decrimento se debido a la introducción de un tercer enlace.

Como ya habíamos deducido antes, no es tanto el efecto de separación de canales lo que perjudica el rendimiento, si no más bien la separación física entre elementos que intervienen en la prueba. Aquí al tener 3 saltos hay una distancia física menor entre los dispositivos, por lo que es normal que se vea algo degradada la calidad de los enlaces. Vemos en la figura 4.13 que lo esperado en el caso de canales no solapados sería 22 Mbps (dividiendo el rendimiento en dos), en nuestro caso tenemos 19 Mbps.

Para el caso de canales solapados, se nos juntan dos agravantes, el hecho de que los canales se interfieran entre sí, y además que las distancias sean menores que para el caso de la prueba 4.5. Por estos motivos la calidad del enlace es bastante pobre.

Observamos por tanto que la mejor opción en esta topología de 3 multisaltos inalámbricos es, como ya esperábamos, poner canales totalmente independientes y que no interfieran unos de otros.

Algoritmo Heurístico

Para comprobar si realmente la suposición que realizamos al realizar el montaje de esta topología, de que la mejor potencia para realizar esto es 16 dBm y cerciorarnos de que la mejor configuración de canales es realmente la de ponerlos separados, pasamos a ejecutar el algoritmo heurístico (figura 4.22).

Lo ideal sería encontrar un resultado muy parecido al obtenido con potencias de 16 dBm, pero configurando potencias menores. Al configurar los enlaces con potencias más bajas, conseguimos que estos interfieran en menor medida con el resto.

Algoritmo para la configuración del escenario de multisalto:

```
for enlace #i de 1 a 7 do
    Configura topología del enlace #i
    for Canales {1,2,3,4,5,6,7,8,9,10,11,12,13} do
    Busca la mejor opción entre los canales.
        for Potencias en {8,10,12,14,16} do
        Busco la mejor opción entre las potencias de la lista
            Guardo el mejor canal y la mejor potencia
        end for
    end for
    Configuramos el enlace #i con el mejor canal y potencia
end for
```

Figura 4.22: Síntesis del Algoritmo Heurístico utilizado en esta prueba

Este algoritmo buscará la mejor configuración de canales y potencias posibles en el escenario de la figura 4.19, con el que obtendremos el rendimiento de la red, para ello realizara pruebas de rendimiento con las combinaciones posibles de parametrización de la red.

Tras la ejecución del algoritmo obtenemos el siguiente resultado:

- **Canal asignado a cada salto:** {5, 13 ,9} CH
- **Potencia asignada a cada salto:** {16, 16, 14} dBm
- **Rendimiento obtenido en cada salto:** {19.1, 18.6, 19.1} Mbps

Vemos como efectivamente la suposición que realizamos al principio para realizar el montaje de 3 multisaltos es muy similar al resultado obtenido en el rendimiento de esta prueba. Tenemos 3 canales no interferentes entre sí (se encuentran al límite pero no están solapados), las potencias son 16 dBm en los dos primeros saltos y 14 dBm en el último salto, esta variación es insignificante ya que entre estos dos valores apenas hay diferencias en las pruebas realizadas anteriormente (sección 4.4).

Comprobamos también con esto que el algoritmo funciona correctamente ya que nos da una configuración bastante buena de los parámetros de la red.

Ahora para llevar un poco más al límite la estabilidad de la red de pruebas mallada multisalto, iremos introduciendo progresivamente un enlace más y cada vez ejecutaremos el algoritmo, con ello comprobaremos como se va comportando la red en cuanto a calidad de la misma, y en que medida va perdiendo calidad al ir añadiendo enlaces hasta llegar al tope de 7 multisaltos, que es la configuración final de la figura 4.20.

Para intentar reducir en cierta medida el efecto de solapamiento de canales, vamos a intentar disminuir la potencia de transmisión en los enlaces, es inmediato pensar que si configuramos una potencia más baja en un enlace, en principio debería molestar menos al resto y obtener una menor interferencia. Para ello hemos incluido en el algoritmo una opción, en la que premiamos a potencias de transmisión más bajas, siempre y cuando no obtengamos una mejora de al menos el 5 % del rendimiento, respecto a la de la potencia inmediatamente superior.

Los resultados obtenidos de esta prueba han sido expuestos en la tabla 4.2:

Resultados de las distintas repeticiones del algoritmo				
Repetición	**Número de Enlaces**	**Canales**	**Potencias (dBm)**	**Rendimiento (Mbps)**
1	3	5	16	19.1
		13	16	18.6
		9	14	19.1
2	4	2	16	19.6
		10	16	20.1
		6	14	19.5
		3	14	12.9
3	5	13	16	20.2
		9	8	19.9
		5	16	19.2
		1	16	12.2
		13	12	12.1
4	6	13	16	19.4
		8	10	19.9
		5	16	19.2
		1	16	12
		13	16	12.6
		9	16	11.3
5	7	13	16	19.4
		8	16	19.9
		5	16	19.2
		1	16	12.5
		13	12	12.4
		9	16	11
		1	14	10.4

Tabla 4.2: Resultados de las repeticiones del algoritmo heurístico incrementando el número de enlaces

Se ve claramente, como mostramos de una manera más detallada en la figura 4.23, que según aumentamos los enlaces en la red para ampliar el número de multisaltos, se reduce el rendimiento de la misma. Esto ocurre desde que introducimos el cuarto salto, ya que como explicamos antes, el número máximo de canales no solapados que pueden estar funcionando a la vez para este modo 802.11g son 3, en este caso por tanto existe solapamiento de canales e interferencia entre ellos.

Vemos que el rendimiento se mantiene constante, en torno a los 19 Mbps, hasta llegar a los 3 saltos, a partir de aquí no queda más remedio que utilizar canales parcialmente solapados para seguir aumentando los enlaces de la red.

Esto repercute en tener una bajada de casi el 50 % del rendimiento, el cual va menguando progresivamente al introducir cada vez más enlaces.

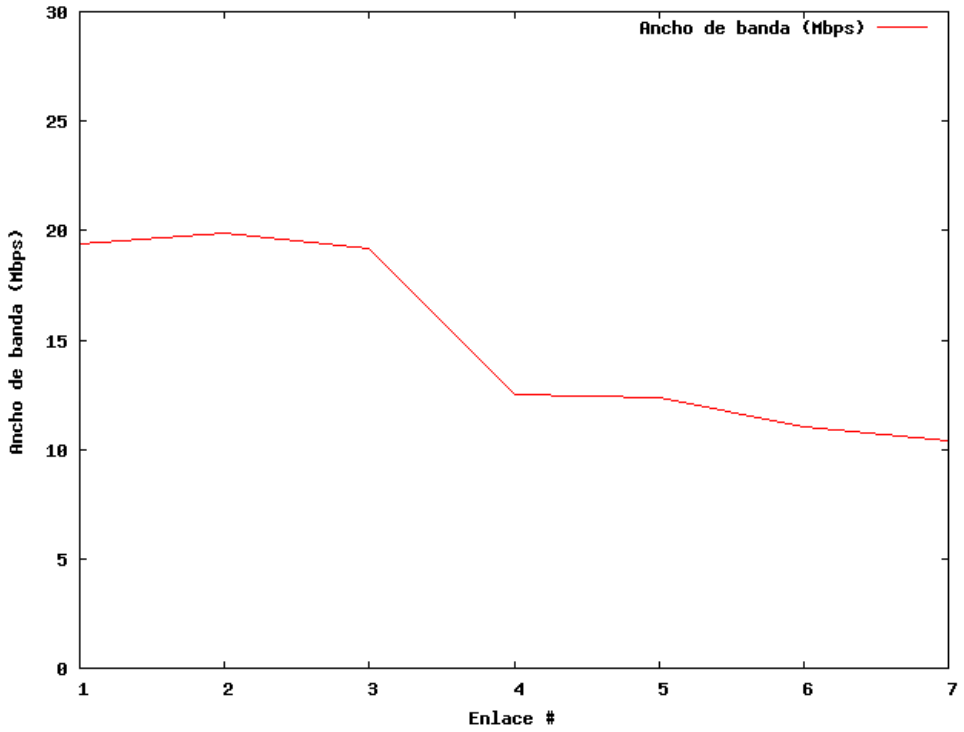


Figura 4.23: Rendimiento en función del número de saltos de la red

Como vemos, a medida que vamos introduciendo enlaces el rendimiento de la red se va viendo afectado, pero es curioso el caso en el que teniendo 4 enlaces, introducimos uno más, aquí el rendimiento apenas sufre cambios manteniéndose casi constante en la mayoría de las repeticiones. A partir de aquí el rendimiento continúa reduciendo su capacidad hasta llegar a los 10 Mbps cuando configuramos la red con el número máximo de saltos.

El caso del cuarto y quinto enlace pensamos que se debe a la ubicación física de los elementos, como vemos en la figura 4.23, el cuarto enlace se compone de los elementos CMP104 y CMP108 que se encuentran bastante separados físicamente de los elemento CMP112 y CMP114 que componen el quinto salto.

Una vez más, observamos con este caso, lo que hemos deducido en pruebas anteriores, el hecho de que afecte más a la ubicación física de los elemen-

tos que el canal utilizado, pese a si este se encuentra solapado parcialmente o no.

Si observamos los parámetros con los que se configura la red en el resultado del algoritmo, vemos que hay pequeñas variaciones en cuanto al canal y a la potencia utilizada, pensamos que esto se debe a que estamos ajustando los valores dentro de unos márgenes muy próximos entre ellos.

En definitiva, la red se comporta dentro de lo esperado, intentando mantener siempre el máximo rendimiento, a pesar de que le hemos puesto la limitación del solapamiento de canales.

Respecto a la potencia parece que en muy pocos casos el algoritmo a optado por configurar una potencia baja frente a una alta, pese a la ventaja que una tenía respecto a la otra. En casi todos los casos se ha optado por la potencia por defecto de 16 dBm, por esto no parece que sea muy crítico, a la hora de configurar la red, el elegir una potencia u otra.

Capítulo 5

Conclusiones y trabajos futuros

A continuación enumeraremos las conclusiones más importantes obtenidas a lo largo de la realización del proyecto. Además incluiremos unas orientaciones que puedan servir para continuar trabajos futuros en la línea de investigación de este proyecto fin de carrera.

5.1 Conclusiones

Hemos dividido en dos partes las principales conclusiones; por un lado tenemos las lecciones aprendidas en lo que al diseño y desarrollo de la plataforma de pruebas se refiere, y por otro, todo lo relacionado con las pruebas de experimentación.

5.1.1 Red de pruebas

Es posible instalar y manejar una plataforma de pruebas bajo el suelo de un laboratorio. Realizando la instalación adecuada, es posible desplegar la red bajo el suelo del laboratorio. Con las ventajas que este diseño conlleva, por ejemplo previene de desconexiones o extravíos inesperados, además de no molestar con el cableado ya que no se ven físicamente signos de la red por ningún sitio.

A la red de pruebas le llegan menos interferencias si esta bajo el falso suelo. Se demuestra en la prueba 4.2, que al tener la red bajo las losetas ais-

lantes del falso suelo, este nos proporciona una protección, y hace de escudo frente a interferencias de otras redes del exterior, en particular de otras redes radiando en la banda de los 2.4 Ghz. Gracias a esto mejoramos el rendimiento en un 10 % respecto a si tuvieramos la red desplegada en el exterior.

El uso de un hub para interconectar los elementos de un mismo nodo afecta al rendimiento de la red. Primeramente se optó por un diseño en el que los elementos de cada nodo de la red estaban conectados a un pequeño hub de 8 puertos, y este a su vez se conectaba con el concentrador principal. Enseguida nos dimos cuenta de que esta manera de conectar la red afectaba al rendimiento de la misma, por lo que se prescindió de este hub y conectamos los elementos directamente al concentrador principal.

La ubicación de los nodos es algo decisivo. La situación de las antenas, y en particular las distancias ente ellas, han de ser mayores que el umbral de campo lejano para así evitar los indeseables e impredecibles efectos de campo cercano.

No todos los firmwares funcionan correctamente en el modelo de Fonera 2100. Al instalar varios firmwares basados en OpenWrt y configurar la Fonera en modo Ad-Hoc, la carga de CPU de esta empieza a aumentar considerablemente, lo que hace imposible su gestión y mucho menos su uso para experimentos. Encontramos el firmware DD-WRT v24 RC 6.2 para tarjetas Atheros WiSoc, con el que la Fonera si permite su manejo a pesar de estar en modo Ad-Hoc.

El uso del router Fonera 2100 para despliegue de redes mesh resulta ineficaz. Tras las primeras pruebas nos dimos cuenta que el rendimiento obtenido con este modelo de router era muy bajo, hasta un 65 % menor que con los otros modelos de enrutadores. Decidimos por tanto no continuar la experimentación con Fonerras.

5.1.2 Conclusiones experimentales

Obtenemos peor rendimiento al generar e inyectar tráfrico en la red con los routers. La diferencia entre generar tráfico con los PC y con los routers es notable en los experimentos realizados, sobre todo funcionando en el

modo 802.11a. El modo 802.11g no nota tanto esta diferencia si no generamos tramas con tamaño menor al de referencia 1500 bytes.

Utilizar la red a ciertas horas del día y en ciertos canales ayuda a aumentar el rendimiento. Durante la experimentación vimos que las horas centrales del día, entre las 9 AM. y las 8 PM., hay algunos canales que se encuentra ligeramente más ocupados, por lo que decidimos realizar, en la medida de lo posible, el resto de experimentos en horario nocturno.

Variando la potencia de transmisión podemos conseguir diversas topologías de red multisalto. En nuestra red disponemos de 14 nodos, por lo que disponemos de hasta 182 posibles enlaces entre ellos. El utilizar una potencia baja, del entorno de 4 dBm, nos permitiría configurar unos 40 enlaces, pero todos ellos con un rendimiento mediocre. En cambio si aumentamos la potencia hasta valores altos del entorno de 16 dBm podemos utilizar más de 100 enlaces diferentes y todos ellos sin bajar de 20 Mbps el rendimiento.

La interferencia entre canales de dos enlaces de la red es algo a tener en cuenta. Dos canales adyacentes o solapados en frecuencia, provoca que el rendimiento de la red se vea afectado. Esto ocurre sobre todo si la interferencia nos llega desde un punto cercano. Como ya dijimos antes la ubicación de los nodos es algo decisivo.

El utilizar un algoritmo heurístico para configurar la red multisalto nos permite disponer una configuración óptima. El algoritmo que se diseñó para este PFC, permite obtener la mejor combinación de potencias y canales en la red. No siempre configurar al máximo la potencia de los enlaces es lo más óptimo.

Configurar una topología multisalto de más de 3 enlaces inalámbricos en el modo 802.11g afecta al rendimiento de la red. Al tener que utilizar más de tres enlaces estamos obligados a usar canales parcialmente solapados, con lo que el rendimiento bajará a partir de este tercer salto. Esto se debe a la escasez de espacio disponible para el despliegue de la red, en un entorno más extenso se podrían separar los nodos bastante más de lo que se ha hecho en este proyecto y podría usarse la reutilización espacial para llevar a cabo un despliegue con un mayor número de saltos.

5.2 Trabajos futuros

En este proyecto, hemos pretendido caracterizar de una manera genérica esta plataforma de pruebas basada en multisaltos inalámbricos. Siguiendo en la línea de investigación y experimentación de este proyecto, se puede ir más allá y realizar experimentos más concretos pensados para un tipo de tráfico específico.

Por ejemplo se podría estudiar el comportamiento de la plataforma de pruebas al inyectar en esta cualquier tipo de tráfico multimedia. En particular sería interesante ver el comportamiento de la red al transportando tráfico de voz IP a través de los multisaltos.

Otro posible estudio que se puede realizar consistiría en combinar los distintos modelos de routers para llevar a cabo los multisaltos en la red mesh, y ver si las prestaciones analizadas en este proyecto se ven afectadas de alguna manera.

Se podrían realizar pruebas de protocolos y algoritmos varios, como por ejemplo de enrutado automático, QoS o autoconfiguración de los elementos, utilizando la red desplegada.

Sería interesante instalar un sistema que permita reiniciar los routers en remoto (APCs), ya que uno de los inconvenientes que nos hemos encontrado a lo largo de este proyecto, a sido la molestia de tener que estar físicamente a la hora de reiniciar los dispositivos en algunas ocasiones que estos no respondian.

Como ya se comentó, este proyecto fin de carrera queda enmarcado dentro del proyecto europeo CARMEN. La plataforma que hemos desplegado se utilizará en el futuro para la evaluación del rendimiento de parte de los componentes desarrollados en el proyecto CARMEN.

Ante cualquiera de las posibilidades comentadas o cualquier otra que los interesados propongan, la plataforma de pruebas queda a disposición de nuevos proyectos, experimentos, estudios o temas relacionados con redes mesh. Además existe una web donde se recoje todo el material relacionado con esta plataforma de pruebas[1].

[1] www.floornet.org

Parte IV

Apéndices

Apéndice A

Presupuestos y diagrama de tareas

A.1 Introducción

Realizamos a continuación, una valoración económica de todo el proyecto. Se incluye en el, las primeras partes del proyecto en las que se realizaron tareas de diseño, la fase de desarrollo e instalación de la red de pruebas, y la parte de experimentación.

Tenemos en cuenta los bienes tangibles que nos han sido necesarios para la realización de la red, como los routers, el cableado y el resto de equipos de la instalación. Además hay que tener en cuenta el trabajo realizado por personal técnico, tanto en la instalación de la plataforma de pruebas como en la realización de los experimentos.

En este caso no ha sido necesario subcontratar ninguna tarea por parte de empresas externas, por lo que este apartado no aplica en el desglose del presupuesto.

En el diagrama de Gantt disponemos de una forma ordenada las tareas, además se asigna a cada una de ellas los recursos humanos necesarios para su realización.

A.2 Presupuesto del Proyecto

UNIVERSIDAD CARLOS III DE MADRID
Escuela Politécnica Superior

PRESUPUESTO DE PROYECTO

1.- Autor:
Miguel Angel Flores Trueba
2.- Departamento:
Ingeniería Telemática
3.- Descripción del Proyecto:
- Titulo **Despliegue y experimentación en una red Mesh 802.11**
- Duración (meses) **3,5 Meses**
Tasa de costes Indirectos: 20%

4.- Presupuesto total del Proyecto (valores en Euros):
17.272 Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar - solo a titulo informativo)	Categoría	Dedicación (hombres mes) a)	Coste hombre mes	Coste (Euro)	Firma de conformidad
					0,00	
Ingeniero A		Ingeniero Senior	1,1	4.289,54	4.718,49	
Ingeniero B		Ingeniero	3,5	2.694,39	9.430,37	
					0,00	
					0,00	
			Hombres mes 4,6	Total	14.148,86	

a) 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable a)
Cableado Eléctrico y regletas eléctricas	130,00	50	3,5	60	3,79
Cableado de datos y conectores	95,00	100	3,5	60	5,54
Ordenadores de sobremesa	750,00	30	3,5	60	13,13
Switches D-Link 24 puertos	250,00	100	3,5	60	14,58
Linksys Wrt54GL	728,00	100	3,5	60	42,47
Asus WL-500 GP	1.050,00	100	3,5	60	61,25
Antena Asus WL-ANT 168	308,00	100	3,5	60	17,97
Tarjeta Atheros para IEEE 802.11a	504	100	3,5	60	29,40
Fonera 2100	280,00	100	3,5	60	16,33
				Total	204,46

a) Fórmula de cálculo de la Amortización:

$$\frac{A}{B} x C x D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado
B = periodo de depreciación (60 meses)
C = coste del equipo (sin IVA)
D = % del uso que se dedica al proyecto (habitualmente 100%)

SUBCONTRATACIÓN DE TAREAS

Descripción	Empresa	Coste imputable
NO APLICA	NO APLICA	NO APLICA
	Total	0,00

OTROS COSTES DIRECTOS DEL PROYECTO a)

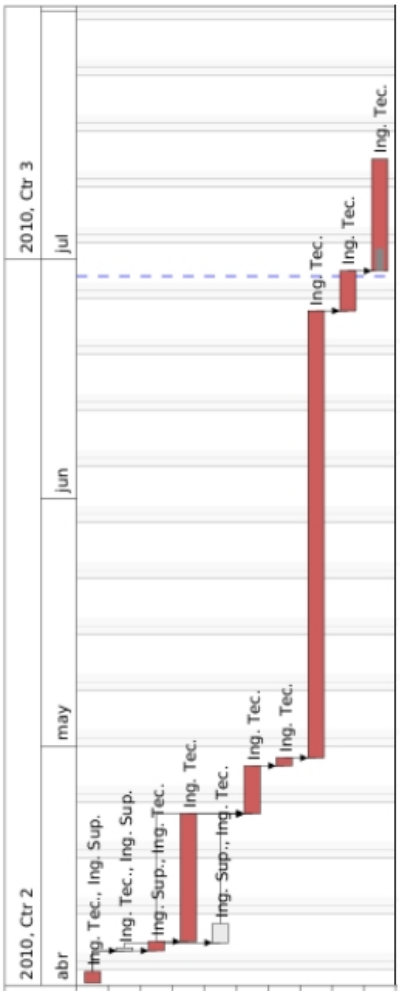
Descripción	Empresa	Costes imputable
Material de oficina (folios, tinta)		30,00
Ventosa manual		10,00
	Total	40,00

a) Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...

6.- Resumen de costes

Presupuesto Costes Totales	Presupuesto Costes Totales
Personal	14.149
Amortización	204
Subcontratación de tareas	0
Costes de funcionamiento	40
Costes Indirectos	2.879
Total	17.272

A.3 Diagrama de Gantt



Tareas

WBS	Nombre	Inicio	Fin	Trabajo	Prioridad	Terminado	Coste	Notes
1	Diseño: Fase 1	Apr 1	Apr 2	4d		0%		Estudio técnico del entorno, elección de elementos y software
2	Diseño: Fase2	Apr 5	Apr 5	2d		0%		Ubicación de los elementos de la red
3	Instalación software y herramientas	Apr 5	Apr 6	3d		0%		Instalación de software y herramientas de analisis, en los elementos de la red
4	Modificación de firmwares en routers	Apr 6	Apr 22	12d		0%		Probar y configurar los firmwares para los distintos routers
5	Primer despliegue	Apr 6	Apr 8	6d		0%		Instalación del cableado eléctrico y cableado de datos. Instalar concentradores
6	Despliegue final	Apr 22	Apr 28	4d		0%		Instalación final de los equipos bajo el falso suelo.
7	Primeras pruebas de conectividad con dispositivos	Apr 28	Apr 29	1d		0%		Pruebas de conectividad con dispositivos , ping, y reset en los equipos.
8	Pruebas experimentales	Apr 29	Jun 24	40d		0%		Experimentos para caracterizar la plataforma de pruebas
9	Recopilación de datos y conclusiones	Jun 24	Jun 29	3d		0%		Recopilación de resultados de experimentos y gráficas.
10	Redacción de la memoria del proyecto	Jun 29	Jul 13	10d		20%		Redactar la memoria del proyecto.

Recursos

Nombre	Nombre corto	Tipo	Grupo	Correo electrónico	Coste
Ingeniero Superior de Telecomunicaciones	Ing. Sup.	Trabajo			0
Ingeniero Técnico de Telecomunicaciones	Ing. Tec.	Trabajo			0

Coste final del Proyecto

El presupuesto total de este proyecto asciende a la cantidad de 17.272 €

Leganés, a 18 de Julio de 2010

El ingeniero proyectista

Apéndice B

Anexos

B.1 Anexo: Cómo Instalar OpenWrt en un router ASUS WL-500g Premium

Pasos previos:

Antes que nada, el PC desde el que trabajemos debe disponer de una tarjeta de red Ethernet configurada con una dirección IP de este tipo: 192.168.1.10/24, introduciendo el siguiente comando en una consola de comandos en linux conseguimos configurar esto:

```
#sudo ip addr add 192.168.1.10/24 dev eth0
```

Una vez realizados estos cambios comenzamos de la siguiente manera:

A- Poner el router en modo Diálogo (diag mode):

Para instalar el OpenWrt usando TFTP (Trivial File Transfer Protocol) o la herramienta de restauración de firmaware de Asus, hay que poner el router en modo dialogo (diag mode). Para ello seguimos estos pasos:

1. Desenchufar el router de la red eléctrica.
2. Asegurarse que el PC esta configurado vía DHCP.
3. Conectar el puerto LAN1 del router al PC. (cable Ethernet).
4. Pulsar el botón negro: RESTORE utilizando un lapicero, y mantener pulsado el botón.

5. Conectar el router a la red eléctrica, a la vez que mantenemos pulsado el botón RESTORE durante unos segundos.

6. Cuando la luz de power parpadee lentamente, querrá decir que ya hemos configurado el modo diálogo.

7. Ahora el router debería aceptar el uso de TFTP.

B- Actualizar el firmware por medio de TFTP:

TFTP (Trivial File Transfer Protocol): Utiliza el puerto 69 en UDP, este protocolo es un protocolo simple, de paso a paso regulado, para la transferencia de archivos que permite a un cliente leer o escribir un archivo en un servidor remoto. Toda la información técnica acerca de este protocolo se encuentra en el RFC 1350 .

Paso para conectar el router mediante TFTP, desde el terminal escribimos:

```
#tftp
#connect 192.168.1.1
```

Pasándole el carácter: ? vemos un listado de comandos.

Ahora vamos a actualizar el router con la ultima versión del firmware OpenWrt, en este caso será "openwrt-brcm-2.4-squashfs.trx"[1]:

```
#tftp> binary
#tftp> trace
#tftp> put openwrt-brcm-2.4-squashfs.trx
```

Después de varias lineas de envíos y confirmaciones, finalmente debería ponernos algo como:

```
#tftp > Sent 1839104 bytes in 7,8 seconds
```

Como referencia tomaremos el tamaño del archivo, debe ser exactamente 1839104bytes. Hay que tener en cuenta que el fichero "openwrtbrcm2.4-squashfs.trx", debe encontrarse en el directorio del PC desde el cual lanzamos el comando tftp.

[1]obtenido en: http://downloads.openwrt.org/kamikaze/7.09/brcm-2.4/ [5/6/2010]

[1]Importante: Después de actualizar el router mediante el put, hay que dejar unos seis minutos de descanso, debido a que el firmware primero es cargado en la memoria RAM y después es flasheado, este proceso tarda aproximadamente **seis minutos**. Tras haber esperado este periodo de tiempo, el router deberá reiniciarse automáticamente, en caso contrario, esperaremos un poco más y lo reiniciaremos manualmente quitando el cable de alimentación y volviéndolo a poner

C- Actualizar la configuracion de red del router:

Hacer telnet 192.168.1.1 con el puerto LAN1 del router conectado al PC, si esta correctamente instalado el nuevo firmware nos dará una pantalla de bienvenida.

```
BusyBox v1.4.2 (2007-09-29 09:01:24 CEST) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 KAMIKAZE (7.09) ----------------------------------
  * 10 oz Vodka       Shake well with ice and strain
  * 10 oz Triple sec  mixture into 10 shot glasses.
  * 10 oz lime juice  Salute!
 --------------------------------------------------
```

Configurando VLANs y rompiendo el bridge entre LAN y WLAN

Como hemos dicho antes, este modelo de router Asus WL-500g Premium, tiene 5 puertos Ethernet, estos puertos no deberían de intercambiar datos entre sí, al menos no todos ellos, y necesitamos separarlos de manera lógica. Para conseguir esto configuraremos los puertos en distintas VLANs, el puerto 1 pertenecerá a la VLAN1 (dispositivo eht0.0), los puertos 2, 3 y 4 se encontrarán separados en la VLAN2 (dispositivo eth0.1) Además el router, lleva por defecto un puente hecho entre el interfaz de la red inalámbrica y los 5 puertos (ver figura B.1) de la red cableada Ethernet. Como vamos a utilizar los puertos Ethernet para la tarea de gestión y el interfaz inalámbrico para enviar y recibir datos, sería conveniente que quedaran separados. Para ello debemos ir a la ruta adecuada, editar los siguientes ficheros y dejarlos de la siguiente manera:

- Fichero de Configuración de red:

```
#vim /etc/config/network

#### VLAN configuration
# WAN > eth0.2 (wan)
# LAN1 > eth0.0 (lan)
```

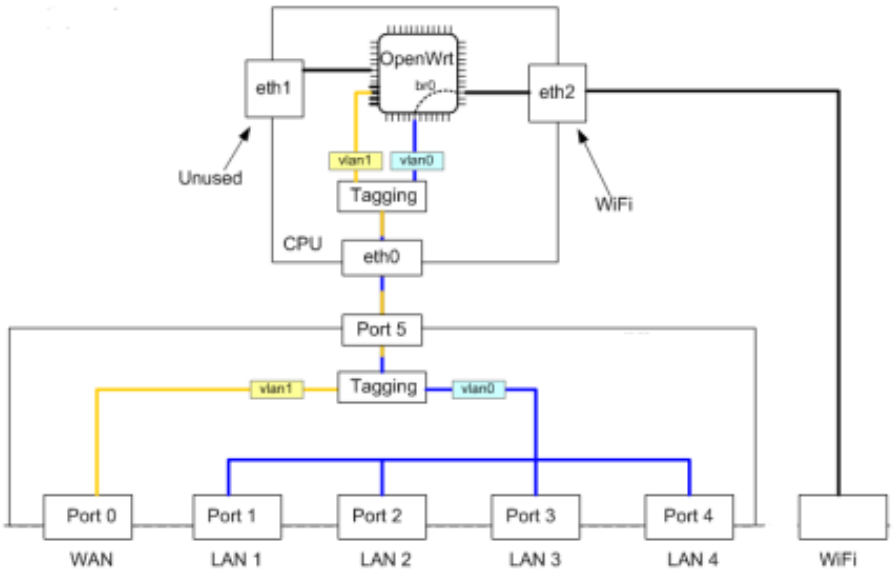


Figura B.1: Puente entre puertos WLAN y LAN

```
# LAN2, LAN3, LAN4 > eth0.1 (lan2)
config switch eth0
        option vlan0    "1 5*"
        option vlan1    "2 3 4 5"
        option vlan2    "0 5"
#### Loopback configuration
config interface loopback
        option ifname   "lo"
        option proto    static
        option ipaddr   127.0.0.1
        option netmask  255.0.0.0
#### LAN configuration
config interface lan
        option ifname   "eth0.0"
        option proto    static
        option ipaddr   192.168.200.101 *variar esta línea.
        option netmask  255.255.255.0
#### LAN2 configuration
config interface lan2
        option ifname   "eth0.1"
        option proto    static
        option ipaddr   192.168.2.1
        option netmask  255.255.255.0
#### WAN configuration
config interface        wan
        option ifname   "eth0.2"
        option proto    dhcp
#### WiFi LAN configuration
config interface wifi
```

```
        option ifname   "ath0"
        option proto    static
        option ipaddr   192.168.3.101 *variar esta línea,
        option netmask  255.255.255.0
```

- Fichero de Configuración del dispositivo Inalámbrico:

```
#vim /etc/config/wireless

config wifidevice  wifi0
        option type     atheros
        option channel  5
        # REMOVE THIS LINE TO ENABLE WIFI:
        #option disabled 1    *variar esta línea.
        #option diversity 0
        #option txantenna 1
        #option rxantenna 1
config wifiiface
        option device   wifi0
        #option network  wifi
        option mode     adhoc
        option ssid     ASUSproyecto      *variar esta línea con el essid deseado
        option encryption none
```

- Además hay que cambiarle el nombre al router de la siguiente manera:

```
#vim /etc/config/system

          option hostname CMP(ultimo numero de IP, 101,102,103...)
```

Finalmente reiniciamos el router:

```
#reboot
```

D- Instalación de paquetes: Vamos a instalar algunos paquetes que nos serán útiles para nuestros propósitos en el proyecto. Lo primero es conectar boca WAN a Internet para poder descargarnos los paquetes e instalarlos. Hacemos telnet 192.168.200.101*(o la dirección que le corresponda a cada router), y vamos instalando uno por uno los siguientes paquetes[2]:

```
          #ipkg update
          #ipkg install ip
          #ipkg install iperf
          #ipkg install wl
          #ipkg install tcpdump
          #ipkg install kmodmadwifi
```

[2]Es importante que lo hagamos manualmente y uno por uno (no copiar y pegar toda la lista), además de seguir el orden en el que se encuentran.

E- Activar SSH sin contraseña con clave publica. Para activar el SSH y poder entrar desde un equipo en concreto, necesitamos hacer lo siguiente:

1. Desde el equipo de control escribimos:

```
#sshkeygen t dsa (después le damos a enter 3 veces).
```

2. Acceder al router por telnet y activar el ssh, introduciendo el comando: password (ponemos prueba como contraseña).

3. Desde el pc de control, llevamos la clave al router:

```
#scp p /root/.ssh/id_dsa.pub root@192.168.200.x:/tmp/
```

4. Ahora accedemos al router por ssh (recuerda contraseña: prueba)

5. En el router ponemos:

```
#cd /etc/dropbear
```

6. A continuacion escribimos:

```
#cat /tmp/id_dsa.pub >> authorized_keys
```

7. Finalmente:

```
#chmod 0600 authorized_keys

root@OpenWrt:~# ls l /etc/|grep dropbear
drwx    1 root     root            0 Feb 28 15:26 dropbear

root@OpenWrt:~# ls l /etc/dropbear/|grep authorized
rw    1 root     root          626 Feb 28 15:31 authorized_keys
```

Si no están de la forma indicada arriba los modificamos con los siguientes comandos:

```
#chmod 0700 /etc/dropbear
#chmod 0600 /etc/dropbear/authorized_keys
```

Llegados a este punto, tendremos en router totalmente configurado, y listo para ser instalado en la red.

Posible problema de cambio de identidad de un host : A veces, es posible que al conectar a otra máquina aparezca un mensaje de advertencia como el siguiente y no sea posible realizar la conexión:

[2]El asentimiento que tendremos al instalar correctamente cada paquete será Done, de no ser así querrá decir que no se ha instalado correctamente el paquete.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (maninthemiddle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
da:79:52:06:64:23:e6:64:55:5e:f2:d3:d9:fc:b2:47.
Please contact your system administrator.
Add correct host key in /home/mesh/.ssh/known_hosts to get rid of this message.
Offending key in /home/mesh/.ssh/known_hosts:15
RSA host key for 192.168.1.1 has changed and you have requested strict checking.
Host key verification failed.
```

Lo que debemos hacer es editar el siguiente fichero:

```
/home/mesh/.ssh/known_hosts
```

Y borrar la linea que nos haya dicho antes, en el caso del ejemplo la 15.

B.2 Anexo: Cómo instalar OpenWrt en una Fonera (modelo 2100)

Configuración Previa:
Configuramos el interfaz Ethernet con una dirección IP del rango: *169.254.255.X*

```
#sudo ifconfig eth1 169.254.255.10 netmask 255.255.255.0
```

A- Acceso a la Fonera mediante SSH:

Antes de empezar a cambiarle el firmware a la Fonera, debemos modificar unos parámetros con el objetivo de que no se conecte automáticamente a internet y se actualice el firwmare. Para esto seguimos los siguientes pasos:

1. Conectamos la fonera al PC y abrimos un navegador web.
2. Accedemos a la configuración de la fonera desde:

   `http://169.254.255.1`

3. Hacemos click en la opción Avanced/Internet Connection. *user: [admin] password: [admin]*. Y configuramos el modo de IP estática, con los siguientes parámetros (trás introducirlos pulsamos en Submit)[3]:

   ```
   IP: 163.117.140.112 (IP con conexión a Internet.)
   Mask: 255.255.255.0
   Gateway: 163.117.140.2 (Puerta de enlace, la IP del router.)
   DNS Server: 88.198.165.155 (Hack de Kolofonium)
   ```

4. Desconectar la fonera del PC y conectarla a Internet. Hacer click en **Submit** y posteriormente (con un lapicero) presionar el boton de la parte inferior de la fonera para resetearla.

   Esperaremos aproximadamente **15 minutos** a que la fonera se conecte a un servidor radius falso el cual se encarga de enviar los cambios adecuados al router.

5. **Sin apagar** la fonera, desconectamos el cable Ethernet que la conecta al router y la conectaremos al PC. Ahora tenemos un servidor SSH llamado **dropbear** esperándonos en el puerto 22.(El user/pass es root/admin).

   Ese puerto está abierto por defecto en:

   La interfaz WLAN (privada): 192.168.10.1

   La interfaz Zeroconf en la Ethernet: 169.254.255.1

6. Nosotros usaremos la interfaz WLAN, para ello nos tenemos que asociar (conectar) con el SSID llamado MyPlace, para evitarnos problemas lo mejor en este paso es hacerlo gráficamente, en la distribución Linux Ubuntu, si hacemos click en la parte superior derecha a un icono en forma de dos monitores podemos ver las redes Wireless disponibles y conectarnos a la que nos iteresa, en este caso ”MyPlace“.

   Para conectar a este SSID nos pedirá una clave WAP para autenticar, esta se encuentra en una pegatina en la parte inferior de la fonera, a continuacion de S/N:XXXXXXXXXX, son 10 dígitos.

[3]**Nota:** En el ejemplo hemos introducido una dirección IP pública así como la puerta de enlace. Si nuestro entorno es accesible a Internet a través de un router convencional deberíamos poner una IP local del tipo: 192.168.x.x y un gateway que sea la dirección física del propio router.

7. Una vez estamos conectados a la red MyPlace, desde un terminal conectaremos por ssh:

```
#ssh root@192.168.10.1          [password: admin]
```

Trás la autenticación, nos aparecerá una pantalla de bienvenida como la siguiente:

```
BusyBox v1.1.3 (2006.11.21-19:49+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.
 _______   _______   _______
|   ____| |       | |   _   |
|   ____| |   -   | |  | |  |
|   |     |_____ _| |__| |__|
|___|
 Fonera Firmware (Version 0.7.1 rev 1) -------------
  *
  * Based on OpenWrt - http://openwrt.org
  * Powered by FON - http://www.fon.com
 ------------------------------------
root@OpenWrt:~#
```

8. Ya tenemos activado la conexión por ssh en la fonera, si nos surgiera algún problema en los pasos previos podemos resetear la fonera y dejarla con la configuración de fábrica, en el ANEXO B.2.3 se dan unas indicaciones acerca de como realizar este proceso.

B- Haciendo cambios definitivos

Tal y como está la fonera ahora mismo si la apagáramos los cambios no se guardarían, por lo que hay que hacer definitivo estos cambios:

1. Desde el prompt de la fonera editamos el fichero: */bin/thinclient*

```
# vim /bin/thinclient
```

2. En este fichero se encuentra una linea que pone algo así: . */tmp/.thinclient* (al final del fichero), debemos comentarla y dejarla de la siguiente forma:

```
#. /tmp/.thinclient
```

Gracias a esto no se ejecutarán las actualizaciones automáticas.

3. Ahora realizamos un enlace al dropbear en */etc/init.d/* para que siempre se ejecute al arrancar:

```
# ln s /etc/init.d/dropbear /etc/init.d/S50dropbear
```

4. Editamos el archivo: */etc/firewall.user* y descomentamos las siguientes líneas:

```
iptables -t nat -A prerouting_rule -i $WAN -p tcp --dport 22 -j ACCEPT
iptables        -A input_rule     -i $WAN -p tcp --dport 22 -j ACCEPT
```

5. En el mismo fichero de antes incluiremos estas dos lineas para no tener que conectar por la red MyPlace, y poder hacerlo por la LAN normal:

```
iptables -t nat -A prerouting_rule -i $LAN -p tcp --dport 22 -j ACCEPT
iptables        -A input_rule     -i $LAN -p tcp --dport 22 -j ACCEPT
```

6. Ejecutamos el fichero: /etc/firewall.user:

```
# /etc/firewall.user
```

7. Antes de reiniciar la fonera, abrimos un navegador web y accedemos a la dirección `http://192.168.10.1`, después vamos al mismo sitio de antes, para variar la configuracion de red de la fonera, dejándola de esta manera:

```
IP: 192.168.200.2xx (una dirección IP local)
Mask: 255.255.255.0
Gateway: 163.117.140.2 (La IP del equipo de control)
DNS Server: 88.198.165.155 (Hack de Kolofonium)
```

8. Reiniciamos el equipo con reboot:

```
# reboot
```

Esperamos 10 o 15 segundos y podemos comprobar como haciendo ssh de la siguiente manera funciona sin problemas:

Cambiamos la dirección en el PC:

```
# sudo ifconfig eth1 192.168.200.230 netmask 255.255.255.0
```

Accedemos por ssh:

```
# ssh root@192.168.200.2xx      [pass: admin]
```

Ya tenemos activado el ssh fijo en esta fonera.

C-Instalación de OpenWRT en la Fonera:

Para poder trabajar con la Fonera es preciso que instalemos el OpenWRT, para ello seguimos los siguientes pasos:

C.1: Instalando RedBoot:

1. Nos descargamos: `http://ipkg.k1k2.de/hack/openwrtar531x2.4vmlinuxCAMICIA.lzma` en nuestro PC .
2. Se lo pasamos a la fonera por ssh:

   ```
   # scp openwrtar531x2.4vmlinuxCAMICIA.lzma root@192.168.200.2xx:
   ```

3. Accedemos a la fonera por ssh:

   ```
   # ssh root@192.168.200.2xx
   ```

4. Desde la Fonera montamos el fichero que acabamos de pasarle:

   ```
   # mtd e vmlinux.bin.l7 write openwrtar531x2.4vmlinuxCAMICIA.lzma vmlinux.bin.l7
   ```

   Esperaremos unos **minutos** a que termine.

5. Sincronizamos y reiniciamos el equipo:

   ```
   # sync;reboot
   ```

C.2: Actualizando particiones:

1. Descargamos de internet el archivo: `http://fonera.info/camicia/out.hex`
2. Se lo pasamos a la fonera:

   ```
   # scp out.hex root@192.168.200.2xx:
   ```

3. Accedemos a la fonera por ssh y montamos el archivo:

```
# ssh root@192.168.200.2xx
# mtd e 'RedBoot config' write out.hex 'RedBoot config' :
```

4. Sincronizamos y reiniciamos el equipo:

```
# sync;reboot
```

C.3 Paso previo importante: Servidor TFTP.

Necesitamos tener corriendo un servidor TFTP en nuestro PC, para poder servir a la fonera (que será nuestro cliente) unos archivos esenciales para la configuración. Para ver como se monta un servidor TFTP, ver el anexo B.3.

Tenemos que dejar en nuestro servidor TFTP los siguientes archivos:

```
# wget http://downloads.openwrt.org/kamikaze/7.09/atheros2.6/openwrtatheros-
2.6vmlinux.lzma O /srv/tftp/vmlinux
# wget http://downloads.openwrt.org/kamikaze/7.09/atheros2.6/openwrtatheros-
2.6root.squashfs O /srv/tftp/squashfs
```

Comprueba que el servidor TFTP este correctamente instalado y no tenga ningún tipo de restricción, tipo firewall o filtros de alguna clase, hay herramientas como netcat o nmap que vienen bien para comprobar que tenemos abiertos ciertos puertos (Recuerda que el TFTP usa el puerto 69 y funciona con tráfico UDP).

C.4 Accediento a RedBoot:

Antes de empezar con este apartado conviene saber que algunos comandos pueden tardar varios minutos, es muy importante no cortarlos ni apagar la fonera ya que se puede bloquear y dejarla inservible.

1. Cambiamos la IP del PC:

```
# sudo ifconfig eth1 192.168.1.20 netmask 255.255.255.0
```

2. Quitamos la alimentación de la fonera, se la volvemos a poner y a partir de este momento disponemos de **10 segundos** para acceder mediante telnet:

```
# telnet 192.168.1.254 9000
```

Tras conectar nos aparecerán unas lineas como que algo ha ido mal, pero no hay que preocuparse, es normal. Lo que es importante es que tras conectarnos estemos en ”RedBoot“, e inmediatamente pulsemos CTRL+C para evitar que arranque un script. (Si no nos diera tiempo y se lanzara el script, hay que esperar a que termine y empezar desde el paso 2 de este apartado.)par

3. Desde la fonera (RedBoot), tecleamos lo siguiente:

```
> ip_address l 192.168.1.254 h 192.168.1.20
```

4. Ahora tecleamos lo siguiente para transferir el kernel al BootLoader:

```
> load r b %{FREEMEMLO} vmlinux
```

Si nos aparece algo de este estilo, es que vamos bien:

```
RedBoot> load r b %{FREEMEMLO} vmlinux
Using default protocol (TFTP)
Raw file loaded 0x800408000x801007ff, assumed entry at 0x80040800
```

5. Creamos una nueva tabla fis (flash image system):

```
> fis init     [yes]
```

6. Y flasheamos el kernell (operación crítica):

```
> fis create r 0x80041000 e 0x80041000 vmlinux.bin.l7
```

Esperar varios minutos.

7. Necesitamos conocer unas posiciones de memoria concretas, para ello, tecleamos lo siguiente:

```
> fis free
```

Nos aparece algo como: 0xA80F0000 .. 0xA87E0000

8. Ahora debemos restar estos dos números en hexadecimal, desde el PC tecleamos los siguientes comandos (uno a uno):

```
$ bc
obase=16
ibase=16
A87E0000 ? A80F0000
```

9. Lo que nos da como resultado otro número hexadecimal:*6F0000* (apuntar este número para después).
10. Cargamos el gestor de ficheros:

```
> load r b %{FREEMEMLO} squashfs
```

11. Lo flasheamos (operación crítica, aquí debemos usar el número calculado en pasos anteriores):

```
> fis create l 0x6f0000 rootfs
```

Esperar varios minutos

12. Reseteamos el equipo

```
# reset
```

Cerramos la actual ventana de comandos (ya que se quedará bloqueada), y abrimos otra nueva.

D-Últimos ajustes: Ya hemos instalado el OpenWrt, ahora hay que configurarlo como es debido:

1. Hacemos telnet a la fonera

```
# telnet 192.168.1.1
```

Lo que nos debería de dar una pantalla de bienvenida de este tipo:

```
   Connected to 192.168.1.1.
Escape character is '^]'.
 === IMPORTANT ============================
  Use 'passwd' to set your login password
  this will disable telnet and enable SSH

BusyBox v1.4.2 (20070929 07:21:40 CEST) Builtin shell (ash)
Enter 'help' for a list of builtin commands.
  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 KAMIKAZE (7.09) ----------------------------------
  * 10 oz Vodka       Shake well with ice and strain
  * 10 oz Triple sec  mixture into 10 shot glasses.
  * 10 oz lime juice  Salute!
 ---------------------------------------------------
```

2. Activamos ahora el acceso por ssh:

```
# passwd
```

   Escribimos la password que queramos, sin olvidarnos de ella.

3. Variaremos algún parámetro de la configuracion de red y el nombre del equipo:
   Editamos:

```
# vim /etc/config/network
```

   Y variamos la siguiente linea:

```
option ipaddr   192.168.200.2xx (La dirección IP local de la fonera) .
```

   Editamos:

```
# vim /etc/config/system
```

   Y variamos la siguiente linea:

```
option hostname CMP2xx (El nombre correspondiente a la dirección IP)
```

4. Reseteamos el equipo:

```
#reboot
```

E-Activar acceso por SSH sin contraseña, con clave pública.

Para activar el SSH y poder entrar desde un equipo en concreto, necesitamos hacer lo siguiente:

1. Desde el equipo de control escribimos (si ya tenemos una clave creada no es necesario hacer esto):

```
#sshkeygen t dsa
```

   (después le damos a enter 3 veces).

2. Desde el pc de control, llevamos la clave al router:

```
# scp p /home/mesh/.ssh/id_dsa.pub root@192.168.200.x:/tmp/
```

3. Ahora accedemos al router por ssh (recuerda la contraseña de antes):

```
# ssh root@192.168.200.2xx
```

4. Escribimos en la linea de comandos de la fonera estos comandos:

```
# cd /etc/dropbear
# cat /tmp/id_dsa.pub >> authorized_keys
# chmod 0600 authorized_keys
```

5. Finalmente reiniciamos el equipo:

```
# reboot
```

6. . Hacer una prueba:

```
# ssh root@192.168.200.2xx
```

Posible problema de versión de firmware de la fonera:

- Comprobar la versión de firmware de la fonera (Esto se ve en el menú status del portal web del router , por ejemplo 0.7.11 ó 0.7.12).
- La Fonera se actualiza automáticamente a la última versión de firmware en cuanto se conecta a internet, por lo que se debe de mantener desconectado el cable ethernet de la fonera hasta haber concluido este procedimiento.
- Si por un casual, ya se ha conectado a internet, lo cual sería lo lógico se puede resetear para que vuelva a la versión de firmware original con el que venía de fábrica de la siguiente forma:
    1. Desenchufar la fonera de la red eléctrica.
    2. Pulsar en el botón de reset que viene en la parte de abajo de la fonera con un objeto punzante.
    3. Manteniendo pulsado este botón de reset, poner el cable de power a la fonera.
    4. Esperar con el reset pulsado al menos 15 segundos.
    5. Soltar el botón de reset.

6. Esperar varios minutos (aunque el proceso podría durar entre 1 y 2 minutos, aunque por experiencia propia recomiendo esperar al menos 10-15 minutos, pues existe un bug por el cual en circunstancias un tanto especiales la actualización del firmware lleva ese tiempo).
7. Desenchufar y volver a enchufar la fonera.
8. Comprobar de nuevo la versión de la fonera, pues tras el flasheo debería haber vuelto a la versión que venía de fábrica. vuelto a la versión que venía de fábrica.

De todas formas si este procedimiento falla, siempre podemos volver a los valores de fábrica, siguiendo las instrucciones que se adjuntan en el anexo B.2.3 de este proyecto.

B.2.1 Incompatibilidades de firmwares en las foneras:

Tras varias pruebas nos dimos cuenta de que no todos los firmwares funcionan correctamente en este modelo de fonera 2100.

Se empezó instalando, al igual que en los otros routers, un firmware kamikaze 7.09 basado en OpenWrt. El principal problema que conlleva el utilizar este firmware es que, al configurar la fonera en modo Ad-Hoc, empieza a relentizarse y tener una carga de CPU muy alta, por lo que se hace imposible la gestión de la misma, y mucho menos el ponerla a funcionar para nuestras pruebas.

Por este motivo decidimos, prescindir de esta versión de firmware y utilizar una más moderna, concretamente la kamikaze 8.09. Los resultados que obtuvimos fueron análogos a los anteriores, y la fonera no permitía manejo alguno al configurarla en modo Ad-Hoc.

Después de probar con las versiones de kamikaze 8.09.1 y 8.09.2-RC2 y obtener los mismos resultados, decidimos intentarlo con otro tipo de firmware basado en DD-WRT.

Con DD-WRT también existen algunas incompatibilidades, probamos con las versiones V24SP1 y V24preSP2. El problema que tuvimos en este

caso fue tras el flasheo de la fonera, comenzó la carga de todo el sistema de la misma, y al llegar al final se reinició y comenzó de nuevo, este proceso se repetía continuamente, por lo que la fonera estaba en un bucle sin fin de reinicio continuo.

Finalmente, después de muchas otras pruebas, el firmware que funcionó correctamente fue uno que en la web de DD-WRT le clasifican en la sección de obsoletos, pero para nuestros objetivos en el proyecto sirve perfectamente.

La versión a la que nos referimos es la DD-WRT v24 RC 6.2 para tarjetas Atheros WiSoc.

B.2.2 Flasheando la fonera:

B.2.3 Dejar una fonera 2100 con los valores de fábrica

Si por algún motivo nuestra fonera se queda bloqueada, y es imposible de manejar o de dejarse instalar otro firmware, podemos hacer lo siguiente para dejar la configuración que teníamos antes de empezar a probar distintas configuraciones. Para ello hemos de conectar la fonera al puerto serie del ordenador.

Para utilizar dicha conexión serie es necesarios el conversor RS232/TTL para adaptar los voltajes del puerto serie del PC (o del adaptador USB/Serie RS232)a la conexión serie de la fonera (TTL).

En muchos comercios de electrónica es posible adquirir este conversor, si no lo encontramos, podemos construirnos uno a partir de un integrado y algunos condensadores, en el esquema de la figura B.2 se muestra como construirlo.

Una vez tenemos el adaptador de puerto serie tenemos que abrir la fonera y realizar las conexiones tal y como indica en la figura B.3.

A continuación conectamos un programa de comunicaciones tipo terminal (minicom o hyperterminal)con la configuración 9600bps, 8N1, encendemos la fonera, esperamos uno segundos y vemos como se ejecutan los scripts de arranque de la fonera. Nada más ver esto pulsamos Control + C y ya estamos en el Redboot.

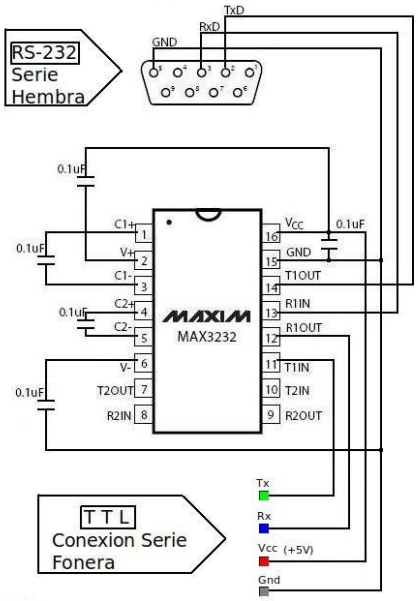


Figura B.2: Esquema conversor puerto serie para fonera

Ahora en nuestro PC debemos configurar un servidor tftpd (ver anexo: B.3) con los dos archivos necesarios para completar el proceso de restauración de la fonera, estos son[4]:

```
rootfs.squashfs
kernel.lzma
```

Volvemos a la ventana del Redboot de la fonera e introducimos los siguientes comandos:

```
ip_address -l 192.168.1.254 -h 192.168.1.xxx
fis init
load -r -v -b 0x80040450 rootfs.squashfs
fis create -b 0x80040450 -f 0xA8030000 -l 0x00700000 -e 0x00000000 rootfs
load -r -b %{FREEMEMLO} kernel.lzma
fis create -r 0x80041000 -e 0x80041000 vmlinux.bin.l7
fis load -l vmlinux.bin.l7
exec
fconfig

Run script at boot: true

Boot script:
>>fis load -l vmlinux.bin.l7
>>exec
```

[4][10/07/2010] Descargar desde http://www.senin.es/fonera/Firmware/fonera.tar.gz

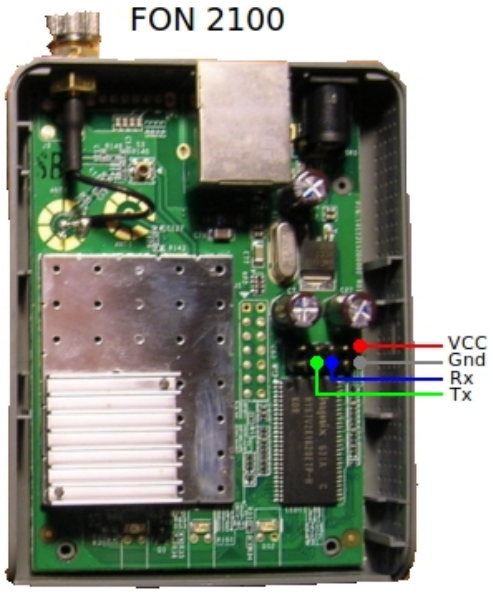


Figura B.3: Conexiones internas de la fonera

```
>> (Si se os queda así pulsad intro un par de veces)

Boot script timeout (1000ms resolution): 10

Use BOOTP for network configuration: false

Gateway IP address:

Local IP address: 192.168.1.254

Local IP address mask: 255.255.255.0

Default server IP address:

Console baud rate: 9600

GDB connection port: 9000

Force console for special debug messages: false

Network debug at boot time: false

Update RedBoot non-volatile configuration - continue (y/n)? y

... Erase from 0xa87e0000-0xa87f0000: .
... Program from 0x80ff0000-0x81000000 at 0xa87e0000: .

RedBoot> reset
```

Después de este reset, la fonera se encontrará con los valores de fábrica del firmware original.

B.3 Anexo: Montar servidor tftpd

En este documento veremos como instalar y configurar un servidor TFTP (Trivial File Transfer Protocol) el cual en nuestro caso será utilizado para proveer la imagen del firmware como medio de respaldo y repositorio de configuraciones de routes, switches y otros equipos de red que soportan TFTP, también es utilizado para proyectos como Linux Terminal Server Project (LTSP).

Hay muchas versiones de aplicaciones para servidores TFTP, en nuestro caso utilizaremos el servidor tftp HPA, y en Debian/Ubuntu lo instalaremos así:

Instalar el paquete:

```
# apt-get install tftpd-hpa
```

Después de instalar el paquete debemos de configurar los parámetros de arranque el demonio tftpd, para ello editamos el archivo de configuración */etc/default/tftpd-hpa*, y lo dejaremos de la siguiente manera:

```
# cat /etc/default/tftpd-hpa
#Defaults for tftpd-hpa
RUN_DAEMON="yes"
OPTIONS="-c -l -s /var/lib/tftpboot"
```

Hemos configurado el directorio */var/lib/tftpboot* como directorio raíz y ahí es donde estarán almacenados los archivos que los clientes tftp descargarán de nuestro servidor.

Debemos cambiar los permisos de escritura de este directorio de esta manera:

```
# chmod 777 /var/lib/tfptboot
```

Por último nos quedará reiniciar el servidor tftpd-hpa, así:

```
# /etc/init.d/tftpd-hpa start
Starting HPA's tftpd: in.tftpd.
```

B.4 Anexo: Herramientas

B.4.1 Manual iperf

Para instalar iperf en los equipos, únicamente debemos teclear lo siguiente:

```
#apt-get install iperf
```

Al tratarse de una herramienta cliente-servidor, tendremos que instalar Iperf como mínimo en dos máquinas. Después se ejecutará iperf en modo cliente o servidor según nos convenga en cada momento.

La forma más básica de ejecución como servidor es:

```
#iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 42.7 KByte (default)
------------------------------------------------------------
```

En este momento Iperf se encuentra a la escucha en el puerto 5001.

En la máquina cliente IPerf, de la forma más sencilla lo ejecutamos de esta manera:

```
#iperf -c 192.168.200.108
------------------------------------------------------------
Client connecting to 192.168.200.108, TCP port 5001
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
```

Conectamos con el servidor (192.168.200.108) y se envían una serie de paquetes para calcular el ancho de banda en la conexión. El resultado es el siguiente:

```
#iperf -c 192.168.200.108
------------------------------------------------------------
Client connecting to 192.168.200.108, TCP port 5001
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  3] local 192.168.200.106 port 1024 connected with 192.168.200.108 port 5001
[  3]  0.0-10.0 sec  60.3 MBytes  50.6 Mbits/sec
```

Vemos que se ha calculado un ancho de banda de 50.6Mbits/sec.

Esto era el manejo básico de Iperf. En la mayoría de nuestras pruebas debemos configurar Iperf de una manera distinta, para ello podemos ejecutar Iperf con los siguientes modificadores:

Como servidor:

A parte de la opción -s que deja a IPref a la escucha, podemos usar:

- -D como demonio.
- -u recibir datagramas UDP en vez de TCP por defecto.
- -P x número de conexiones simultáneas.
- -m muestra MTU (depende del sistema operativo).
- -w especifica el tamaño de Ventana (TCP window size). Muy útil para ir calculando nuestro tamaño de ventana más óptimo según las mediciones de ancho de banda.
- -f[bkmBKB] mostrar resultados en bits/s, kilobits/s, megabytes/s, Bytes/s, KiloBytes/s, MegaBytes/s (s=segundos). Tanto en cliente como servidor.

Como cliente:

Lo más básico es -c IP pero podemos establecer otras opciones, las más importantes:

- -u utilizar datagramas UDP.
- -f [bkmBKB] (igual que lo comentado como servidor).
- -w (lo mismo que para servidor).
- -m muestra MTU (depende del sistema operativo).
- -T ttl especifica valor TTL.
- -i segundos especifica un intervalo, medido en segundos, en el cual se volverá a realizar la medición.
- -t segundos tiempo duración transmisión. Hace más fiable la medida.

Para ver más sobre las distintas opciones de los modificadores que dispone Iperf escribimos:

```
#iperf --help
```

y nos dará un listado de estas opciones con una breve descripción de ellas.

B.4.2 Manual nagios

Los pasos explicados en este anexo han sido probados usando Ubuntu 9.10 - Karmic Koala -, de igual manera podríamos ejecutarlos en otras distribuciones de Linux como Debian.

La mayoría de versiones de Ubuntu vienen con un servidor web apache instalado por defecto, de no ser así deberíamos instalar uno, ya que nagios lo utiliza para funcionar.

Lo primero que tenemos que hacer es instalar nagios, para ello abrimos una ventana de comandos y nos autenticamos como superusuario.

```
#sudo su
```

A continuación, con el equipo conectado a Internet, nos descargamos e instalamos los paquetes de nagios.

```
#apt-get update && apt-get install nagios2
```

No se debe utilizar el apache-config-file que viene con nagios por defecto, ya que en el PC los sitios web se alojan bajo */var/www*.

Lo que hay que hacer es crear el directorio */var/www/nagios*, copiar en él todo el contenido de */usr/share/nagios2/htdocs*, y además este directorio debe tener los permisos para www-data:

```
# mkdir -p /var/www/nagios/htdocs
# cp -R /usr/share/nagios2/htdocs/* /var/www/nagios/htdocs/
# chown -R www-data.www-data /var/www/nagios
```

Ahora se puede crear el fichero */etc/apache2/sites-available/nagios2.conf* con el siguiente contenido:

```
#
# nagios Virtual Host Webinterface
#

<VirtualHost xxx.xxx.xxx.xxx:80>                          ## Tu IP
        ServerAdmin     admin@midominio.com         ## Tu dirección de correo
        DocumentRoot    /var/www/nagios/htdocs

        ServerName      nagios.midominio.com              ## Nombre de dominio

        ErrorLog   /var/log/apache2/nagios.midominio.com-error.log   ## Nombre de dominio
        CustomLog  /var/log/apache2/nagios.midominio.com-access.log combined ## Nombre

        ScriptAlias /cgi-bin/nagios2 /var/www/nagios/cgi-bin
```

```
        ScriptAlias /nagios2/cgi-bin /var/www/nagios/cgi-bin

        <Directory /var/www/nagios>
                Options FollowSymLinks

                DirectoryIndex index.html

                AllowOverride AuthConfig
                Order Allow,Deny
                Allow From All

                AuthName "Nagios Access"
                AuthType Basic
                AuthUserFile /etc/nagios2/htpasswd.users
                require valid-user
        </Directory>

        <Directory "/usr/lib/cgi-bin/nagios2">
                Options ExecCGI

                AllowOverride None
                Order allow,deny
                Allow from all

                AuthName "Nagios Access"
                AuthType Basic
                AuthUserFile /etc/nagios2/htpasswd.users
                Require valid-user
        </Directory>

</VirtualHost>
```

y activamos esta configuración de nagios con:

```
# a2ensite nagios2.conf
```

Para poder acceder a nagios mediante un subdominio del tipo *nagios.midominio.com*, tienes que editar los ficheros:

/etc/apache2/sites-available/ispcp.conf

/etc/ispcp/apache/working/ispcp.conf

Ir a la sección vhost de *midominio.com* y borrar **.midominio.com* de esta linea:

```
ServerAlias     www.midominio.com midominio.com *.midominio.com
```

Después de esto, hemos de recargar la configuración apache:

```
#/etc/init.d/apache2 reload
```

Con esto ya tenemos instalado nagios, ahora vamos a realizar una primera configuración para poder manejarlo:

En el fichero /etc/nagios2/cgi.cfg, cambiar los siguientes datos:

```
physical_html_path=/var/www/nagios/htdocs
url_html_path=/
```

Para darle un poco más de seguridad a nagios, vamos a configurarle una contraseña de la siguiente manera:

```
# htpasswd -c /etc/nagios2/htpasswd.users nagiosadmin
New password:
Re-type new password:
```

Ahora deberías poder acceder a nagios en `http://nagios.midominio.com` y conectarte bajo el usuario nagiosadmin y la contraseña creada en el paso anterior. La configuración de nagios se realiza en */etc/nagios2/*.

Para los propósitos de este proyecto, hemos de configurar nagios de una manera un tanto especial, esto nos ayudará a monitorizar los elementos de la red.

Editamos el directorio */usr/local/nagios/etc/nagios.cfg*, y descomentamos la linea:

```
# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg
```

Con esto Nagios nos permitirá añadir dispositivos a la red para que los monitorice.

Ahora tenemos que modicar las plantillas que vienen en el directorio */usr/local/nagios/etc/objects/switch.cfg* y crear las definiciones de los equipos que nos interesan monitorizar, además del servicio que queremos tener sobre el equipo. En este ejemplo vemos como añadiríamos los equipos que van a ser monitorizados CMP101 y CMP102, además del servicio de ping cada 5 minutos. A continuación de ellos abría que añadir los 40 equipos de la red restantes.

```
###############################################################################
###############################################################################
#
# HOST DEFINITIONS
#
###############################################################################
###############################################################################

# Define the switch that we'll be monitoring
```

```
# Definicion de CMP101
 define host{
        use             generic-switch   ; Inherit default values from a template
        host_name       CMP102 1        ; The name we’re giving to this switch
        alias           ASUS-CMP101    ; A longer name associated with the switch
        address         192.168.200.101         ; IP address of the switch
        hostgroups      switches        ; Host groups this switch is associated
        }
# Definicion de CMP102
 define host{
        use             generic-switch  ; Inherit default values from a template
        host_name       CMP102          ; The name we’re giving to this switch
        alias           ASUS-CMP102    ; A longer name associated with the switch
        address         192.168.200.102         ; IP address of the switch
        hostgroups      switches         ; Host groups this switch is associated
        }
###############################################################################
###############################################################################
#
# SERVICE DEFINITIONS
#
###############################################################################
###############################################################################

# Create a service to PING to switch

# Definicion de servicio PING para CMP101
define service{
        use              generic-service ; Inherit values from a template
        host_name        CMP101  ; The name of the host
        service_description PING            ; The service description
        check_command       check_ping!200.0,20%!600.0,60%;
        normal_check_interval   5   ; Check the service every 5 minutes
        retry_check_interval    1   ; Re-check the service every minute
        }
# Definicion de servicio PING para CMP102

define service{
        use                     generic-service ; Inherit values from a template
        host_name               CMP102  ; The name of the host
        service_description     PING            ; The service description
        check_command      check_ping!200.0,20%!600.0,60%;The command used to monitor
        normal_check_interval   5   ; Check the service every 5 minutes
        retry_check_interval    1   ; Re-check the service every minute
        }
```

B.4.3 Manual tcpdump

Lo primero que debemos averiguar cuando estamos usando el tcpdump, es las interfaces que queremos escuchar. Por defecto cuando se ejecuta sin parámetros, en Linux se pone a escuchar en la eth0. Para averiguar la interfaces en cualquier Unix recurrimos al comando ifconfig -a el cual nos da una lista de las interfaces que tenemos, así como sus parámetros de configuración.

Si queremos escuchar en la interfaz ath0, usaremos tcpdump -i ath0. Cuando estamos leyendo la red, puede que no nos interese que el tcpdump intente resolver los nombres de las maquinas (pueden que no estén dadas de alta en el DNS, por motivos de seguridad, etc), para ello disponemos de la opción -n. Para establecer la longitud de los datos que captura tcpdump usamos -s len, donde len es la longitud que nos interesa. Por defecto el tcpdump sólo captura los primeros 68 bytes, lo cual es útil si lo único que se quiere son las cabeceras IP, TCP o UDP.

Podemos trabajar offline con el tcpdump. Si queremos grabar nuestra captura para posteriormente leerla y analizarla usamos la opción -w file donde file es el nombre del fichero donde queremos grabar la captura de datos. Posteriormente podemos leer y analizar offline con -r file.

Interpretar la salida

Lo primero que hay que decir es que la salida depende del protocolo que estemos analizando. Para empezar comentar que todas las capturas del tcpdump tienen como primer campo una marca de tiempo, que indica cuando ha sido capturado el paquete.

Peticiones ARP/RARP:
Las peticiones arp aparecen de la siguiente manera:

```
18:33:49.908612 arp who-has 192.168.1.2 tell 192.168.1.1
18:33:49.908691 arp reply 192.168.1.2 is-at 0:2:a5:ee:ec:10
```

En este caso, la máquina 192.168.1.1 pregunta por la dirección ethernet 192.168.1.2 (suponemos ambas máquinas en la misma subred). Como vemos

la 192.168.1.2 responde. En este caso, vemos los valores numéricos puesto que se ha usado la opción -n.

Paquetes TCP:

La linea general de un paquete TCP es como sigue:

src > dst: flags [dataseq ack window urgent options]

En principio los parámetros src, dst y flags están siempre presentes. Los otros dependiendo del tipo de conexión TCP que se trate. El significado de dichos parámetros es:

- **src:** Dirección y puerto origen. En caso de no especificar el parámetro -n se intenta resolver el nombre vía DNS y el se busca el nombre del puerto vía (normalmente en los Unix en /etc/services).
- **dst:** Dirección y puerto destino, exactamente igual que el caso anterior.
- **flags:** Indica los flags de la cabecera TCP. Puede ser un ., cuyo significado es que no hay flags, o bien una combinación de S (SYN), F (FIN), P (PUSH), W (reducción de la ventana de congestión), E (ECN eco).
- **dataseq:** El número de secuencia del primer byte de datos en este segmento TCP. El formato es primero:ultimo(n), que significa que desde a primero a ultimo (sin incluir ultimo) hay un total de n bytes de datos. Ojo cuando hay segmentos con SYN, que también ocupa un numero del espacio de secuencia.
- **ack:** El número de asentimiento. Indica el número siguiente de secuencia que se espera recibir. Ojo los SYN también se asienten.
- **win:** Tamaño de la ventana de recepción.
- **urgent:** Existen datos urgentes.
- **options:** Indica la existencia de opciones. En caso de que haya van entre < y >.

En el siguiente ejemplo, (viene en la página web de manual del tcpdump[13]), podemos ver:

```
1: rtsg.1023 > csam.login: S 768512:768512(0) win 4096 <mss 1024>
2: csam.login > rtsg.1023: S 947648:947648(0) ack 768513 win 4096 <mss 1024>
3: rtsg.1023 > csam.login: . ack 1 win 4096
4: rtsg.1023 > csam.login: P 1:2(1) ack 1 win 4096
5: csam.login > rtsg.1023: . ack 2 win 4096
6: rtsg.1023 > csam.login: P 2:21(19) ack 1 win 4096
7: csam.login > rtsg.1023: P 1:2(1) ack 21 win 4077
8: csam.login > rtsg.1023: P 2:3(1) ack 21 win 4077 urg 1
9: csam.login > rtsg.1023: P 3:4(1) ack 21 win 4077 urg 1
```

Esto simula una conexión originada por la máquina rtsg con destino a csam, con el servicio rlogin.

El significado de las líneas anteriores es:

```
1: Inicio de conexión de rtsg -> csam SYN ISN 768512 ventana de 4096
2: SYN de csam -> rtsg ISN 947648 ventana de 4096 ACK del SYN anterior.
3: ACK del SYN mandado por csam. No hay flags
4: 1 byte de datos de rtsg -> csam. Flag PUSH activado., (los números de secuencia
      son relativos al ISN a menos que especifiquemos la opción -S, en cuyo caso pone
      los números de secuencia se imprimen de manera absoluta).
5: ACK del byte de datos anterior por parte de csam.
6: 19 bytes de datos de rtsg a csam.
7: csam manda 1 byte de datos a rtsg, y manda el ACK de los 19 bytes enviados por rtsg.
    La ventana de recepcisn ha bajado en 19 bytes. Flag PUSH.
8: csam envía un byte de datos urgente. Flag PUSH.
9: Igual que el anterior.
```

Datos UDP:

Un paquete UDP se imprime de la siguiente manera:

origen.srcport > destino.dsrpot: udp len

- origen: Nombre o dirección origen.
- srcport: Puerto origen.
- destino: Nombre o dirección destino.
- dstport: Puerto destino.
- len: Longitud de los datos de usuario.

Ejemplo:

12:35:21.457350 10.10.109.10.1025 > 192.168.1.2.1345: udp 121 [ttl 1]

En algunos casos, puede interpretar protocolos que vayan encapsulado en los paquetes UDP, como NFS o DNS. El grado de detalle en la interpretación de estos protocolos dependerá del grado de detalle (controlado con la opción -v) que queramos darle.

Filtros:

Lo más importante que nos permite hacer el tcpdump, es el uso de filtros. Un filtro es una expresión que va detrás de las opciones y que nos permite seleccionar los paquetes que estamos buscando. En ausencia de estos filtros, el tcpdump volcará todo el tráfico que vea el adaptador de red seleccionado.

La expresión que se usa para definir el filtro tiene una serie de primitivas y tres posibles modificadores a las mismas. Esta expresión sera verdadera o falsa y hara que se imprima o no el paquete de datos.

Los 3 modificadores posibles son:

tipo Puede ser host, net o port. indican respectivamente una maquina, por ejemplo host 192.168.1.1 , una red completa, por ejemplo net 192.168, o un puerto concreto, por ejemplo port 22. Por defecto se asume el tipo host.

dir Especifica desde o hacia donde se va a mirar el flujo de datos. Tenemos src o dst y podemos combinarlos con or y and. Para el caso de de protocolos punto a punto podemos sustituir por inbound o outbound.Por ejemplo si queremos la dirección de destino 10.10.10.2 y la de origen 192.168.1.2, el filtro será dst 10.10.10.2 and src 192.168.1.2 . Si se quiere que sea la dirección destino 192.168.1.1 o la dirección origen 192.168.1.2, será dst 192.168.1.1 or src 192.168.1.2. Pueden seguirse combinando con la ayuda de paréntesis o las palabras or y and. Si no existe se supone src or dst. Por supuesto, esto se puede combinar con los modificadores de tipo anteriores.

proto En este caso es el protocolo que queremos capturar. puede ser tcp,udp,ip,ether (en este caso captura tramas a nivel de enlace,arp (peticiones arp), rarp (petciones reverse-arp),fddi(para redes FDDI, pero realmente el encapsulado es igual al ether). Hay otros niveles de enlace para redes Decnet

y lat, pero dado su escaso uso, me remito a la pagina de manual del programa[13].

A continuación se dan las primitivas que pueden usarse. El resto se tiene que poner si queremos poner el filtro con el comportamiento.

* **[dst—src] host maquina**. Cierto si la dirección destino u origen del paquete es máquina lo cual puede ser una dirección IPv4 (o IPv6 si se ha compilado soporte para el mismo), o un nombre del DNS. Si queremos restringir a dirección destino podemos restringir con dst. Para dirección origen src.

Ejemplos:
Capturar el trafico cuya IP origen sea 192.168.1.1

```
tcpdump src host 192.168.1.1
windump src host 192.168.1.1
```

Capturar todo el tráfico cuya dirección origen o destino sea 192.168.1.2

```
tcpdump host 192.168.1.2
windump host 192.168.1.2
```

* **ether src—dst—host edir**. Este filtro es cierto sí la dirección origen (src), la destino (dst) o el cualquiera de las dos(host) coincide con edir. Hacer notar que src,dst o host es obligatorio especificarlo.

Ejemplos:
Capturar el tráfico con destino a la dirección ethernet 0:2:a5:ee:ec:10.

```
tcpdump ether dst 0:2:a5:ee:ec:10
windump ether dst 0:2:a5:ee:ec:10
```

Capturar el tráfico que vaya a la máquina cuya dirección MAC es 0:2:a5:ee:ec:10.

```
tcpdump ether host 0:2:a5:ee:ec:10
windump ethert host 0:2:a5:ee:ec:10
```

* **gateway maquina.** Cierto en caso de que el paquete use maquina como router. maquina debe estar definida en /etc/ethers y /etc/hosts. Realmente los paquetes que cumplen con esa condición son aquellos que tienen como

dirección ethernet destino máquina, pero ni la dirección IP destino u origen es máquina.

*** [dst—src] net red.** Cierto en caso de que la red de la dirección destino, origen o ambas sea red. El parámetro red puede ser una dirección numérica (por ejemplo 192.168.1.0) o bien un nombre que se resuelve a dirección, en los Unix, con ayuda del /etc/networks. Decir que también se admite el clásico direccionamiento CIDR. Podemos especificar una máscara poniendo red como net red mad máscara o bien usar /, net red/bits. Hacer notar que el uso de net... mask no es compatible con direcciones IPv6. Si queremos hacer referencia a la red destino usamos dst como prefijo. Para la red origen usamos dst.

Ejemplos:
Capturar todo el tráfico cuya red destino sea 192.168.1.0.

```
tcpdump dst net 192.168.1.0
windump dst net 192.168.1.0
```

Capturar todo el tráfico cuya red origen sea 192.168.1.0/28

```
tcpdump src net 192.168.1.0 mask 255.255.255.240
tcpdump src net 192.168.1.0/28
```

Capturar todo el tráfico con origen o destino en la 10.0.0.0/24

```
tcpdump net 10.0.0.0/24
tcpdump net 10.0.0.0 mask 255.255.255.0
```

*** [dst—src] port puerto**. Cierto en caso de que el puerto (ya sea udp o tcp) coincida con puerto. Si no se especifica dst o src, será cierto tanto puerto origen como destino. Si queremos restringir a destino usamos dst y a origen usamos src. El puerto es un valor numérico entre 0-65535 o bien un nombre que en Unix se resuelve a través del /etc/services.

Ejemplos:
Capturar todo el tráfico con destino al puerto 23

```
tcpdump dst port 23
```

Capturar todo el tráfico con destino o origen puerto 80

```
tcpdump port 23
```

* **less longitud**. Cierto en caso de que el tamaño del paquete sea menor o igual longitud.

* **greater longitud.** Cierto en caso de que el tamaño del paquete sea mayor o igual que longitud.

* **ip proto protocolo.** En este caso escucha el protocolo que se le indique. El protocolo puede ser icmp, icmp6, igmp (internet group managent protocol), igrp (interior gateway routing protocol), pim (protocol independent multicast), ah (IP Authentication header), esp (encapsulating security payload), udp o tcp. En caso de usar icmp, udp o tcp hay que escapar el protocolo, poniendo un /, es decir, ip proto / icmp. Ojo con ese caracter que también hay que escaparlo en los shells de Unix.

Por comodidad se disponen los alias tcp, udp e icmp que equivalen a ip proto tcp or ip6 proto tcp, etc.

Ejemplos:
Capturar todos los paquetes icmp

```
tcpdump ip proto \\ip
```

Capturar todo el tráfico udp

```
tcpdump ip proto \\udp
tcpdump udp
```

* **ip6 proto protocolo.** Cierto si es un paquete de IPv6 con el protocolo protocolo.

* **ip6 protochain protocolo.** Es un número que en los Unix puede leerse en /etc/protocols. En este caso lo que se busca es que dentro de los diferentes cabeceras que puede tener un paquete IPv6 una de ellas sea el protocolo especificado.

* **ip protochain protocolo**. Igual que el caso anterior pero para IPv4.

* **ether broadcast.** Cierto si la trama capturada va dirigida hacia la dirección de difusión ethernet. La palabra ether es opcional.

* **ip broadcast.** Cierto si el paquete va dirigido a la dirección de difusión de IP. Esta dirección se comprueba si es todo 0 o 1, o bien se comprueba la dirección local de la subred.

* **ether multicast**. Cierto si la trama va dirigida a una dirección multicast ethernet.

* **ip multicast.** Cierto si el paquete va dirigido a una dirección multicast IP.

* **ip6 multicast.** Cierto si el paquete va dirigido a una dirección multicast IPv6.

* **ether proto protocolo.** Cierto si el protocolo que contiene la trama es de tipo protocolo Los protocolos son ip, ip6, arp, rarp, atalk, aarp, decnet, sca, lat, mopdl moprc e iso. Además estos nombres son identificadores que deben de ser escapados con .

Sin embargo hay una serie de alias que hacen mas cómodo la expresión en los filtros. Dichas expresiones son ip,ip6 ,arp,rarp,aarp, decnet e iso, siendo equivalentes a ether proto ip, ether proto ip6, etc.

Ejemplos: o Capturar todo tráfico arp

```
tcpdump -n ether proto \\arp
tcpdump -n arp
```

(el alias es más cómodo)

Capturar todo tráfico ip

```
tcpdump -n ether proto \\ip
tcpdump -n ipi
```

* **vlan [vlanid].** Cierto si la trama capturada es un paquete 802.1Q VLAN. Hacer notar de que esto cambia el resto de la interpretación del paquete capturado, en especial los desplazamientos a partir de los cuales empiezan a decodificar los protocolos, ya que se asume que estamos capturando paquetes que viajan en tramas VLAN. Por último si esta presente el parametro vlanid, sólo se mostraran aquellos paquetes que vayan a la VLAN vlanid.

B.4.4 Otras herramientas y comandos

Para una mayor descripción de estos comandos, ejemplos y modificadores, escribir en cualquier terminal de Linux ”man“ seguido del comando, nos aparecerá el manual siempre y cuando tengamos instalado el programa en el equipo.

ifconfig es un programa disponible en varias versiones del sistema operativo UNIX, que permite configurar o desplegar numerosos parámetros de las interfaces de redes, como la dirección IP (dinámica o estática), o la máscara de red. Si se llama sin argumentos suele mostrar la configuración vigente de las interfaces de red activas, con detalles como la dirección MAC o el tráfico que ha circulado por las mismas hasta el momento.

iwconfig análogo al anterior, pero muestra la configuración de los dispositivos de red inalámbricos.

iwlist se utiliza para mostrar información detallada de una interfaz de red inalámbrica, incluida la información que se ha mostrado de iwconfig.

iwpriv con este comando se le pueden pasar a las tarjetas unos parámetros y con figuración específica de cada interfaz (en contraste con iwconfig que presentaba funcionalidades genéricas).

vi es un programa informático que entra en la categoría de los editores de texto. Esto es así, pues a diferencia de un procesador de texto no ofrece herramientas para determinar visualmente cómo quedará el documento impreso. Es por esto que carece de opciones como centrado o justificación de párrafos, pero permite mover, copiar, eliminar o insertar caracteres con mucha versatilidad. Este tipo de programas es frecuentemente utilizado por programadores para escribir código fuente de software. Existe una versión mejorada que se llama vim.

B.5 Anexo: Diagramas de flujo

B.5.1 Script generar tráfico en PC o en routers

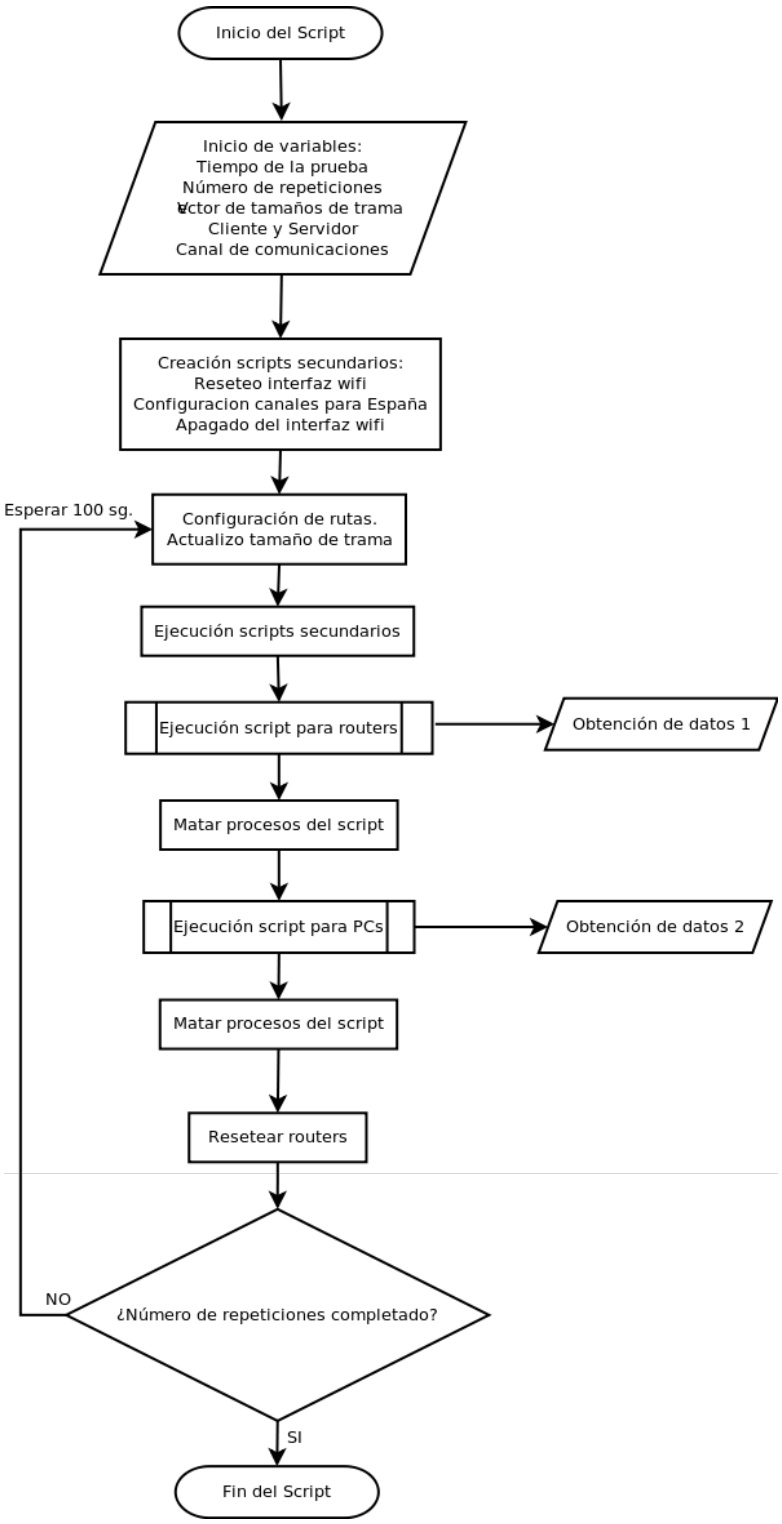


Figura B.4: Diagrama de Flujo de la prueba 4.1

B.5.2 Script del impacto de la hora del día

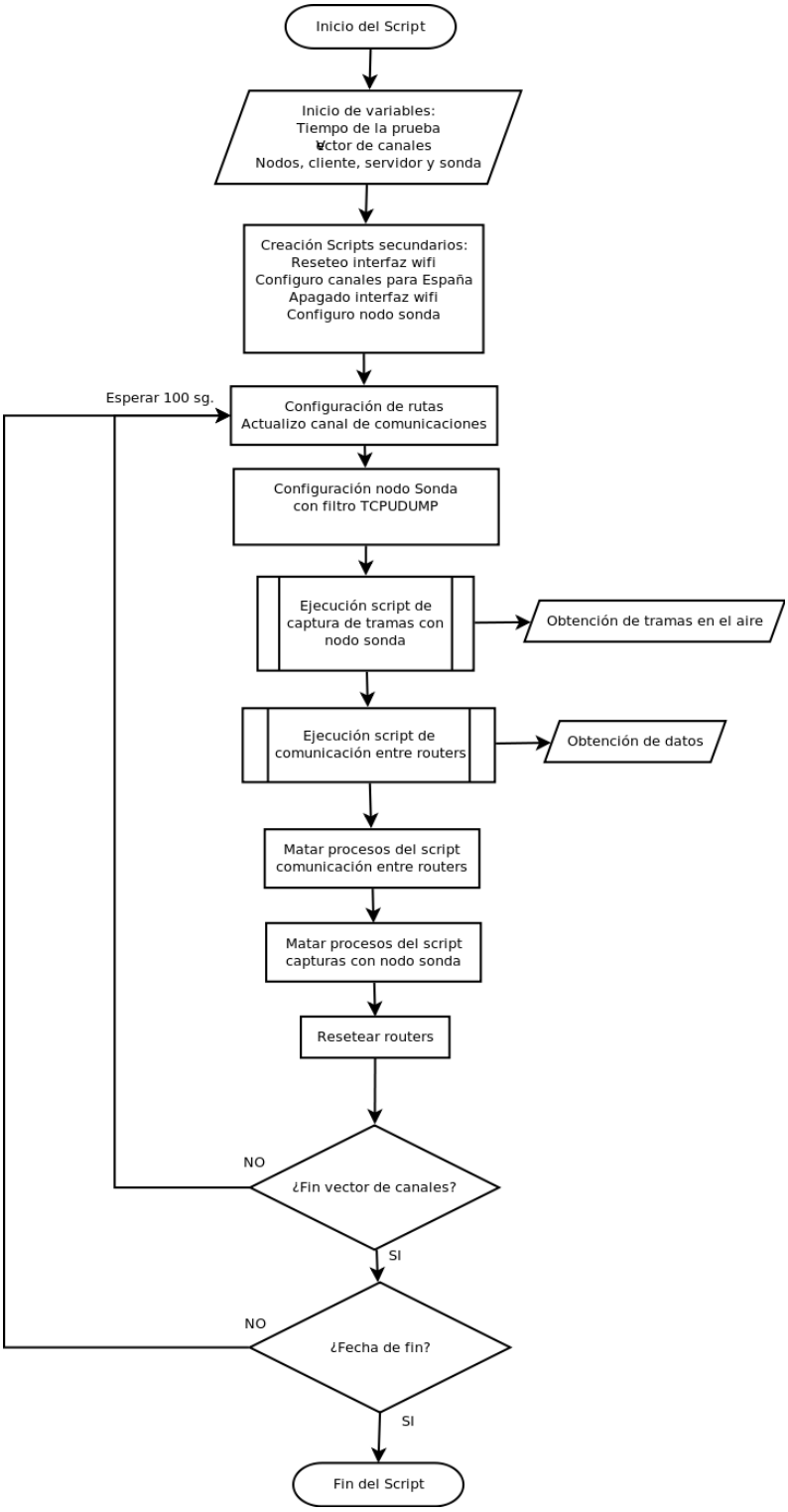


Figura B.5: Diagrama de Flujo de la prueba 4.3

B.5.3 Script de filtrado de canales wifi

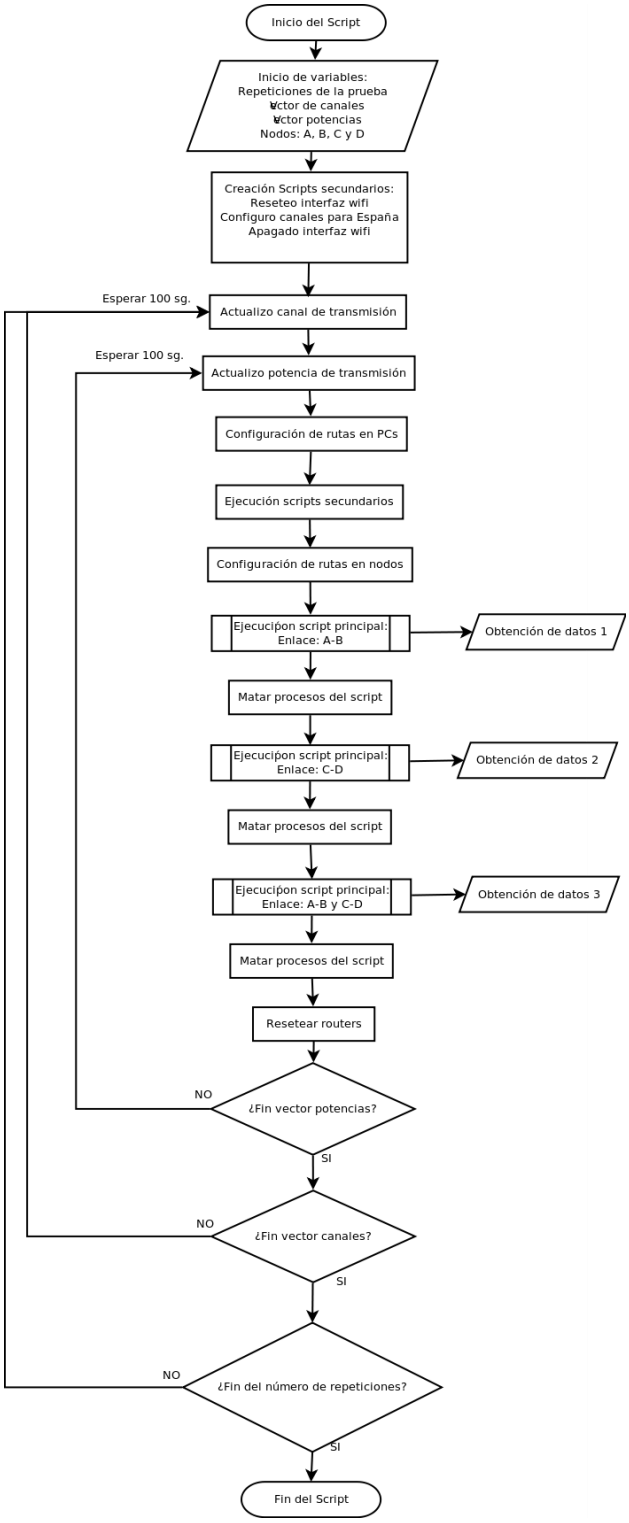


Figura B.6: Diagrama de Flujo de la prueba 4.3

B.6 Ayudas y manuales referenciados en los anexos

- Configurar puerto serie en la fonera:

  http://www.tufonera.com/index.php/Puerto_serie_Fonera_2100_y_2200

- Foro de información sobre firmwares de la fonera:

  http://www.fonera.info/

- Wiki sobre flasheado de foneras:

  http://wiki.openwrt.org/OpenWrtDocs/Hardware/Fon/Fonera

- Como instalar un servidor tftp:

  http://tuxjm.net/2008/11/25/ubuntu_como_instalar_un_servidor_tftp_con_hpa_tftp_server/

- Información acerca de nagios:

  http://nagios.sourceforge.net/docs/2_0/

B.7 Anexo: Tabla de direccionamiento y ubicación de equipos

Direccionamiento y ubicación del equipamiento				
Nodo	**Baldosa**	**Modelo**	**Dirección IP**	**Nombre**
1	B-3	Linksys	192.168.200.1	CMP001
		Asus	192.168.200.101	CMP101
		Fonera	192.168.200.201	CMP201
2	F-3	Linksys	192.168.200.2	CMP002
		Asus	192.168.200.102	CMP102
		Fonera	192.168.200.202	CMP202
3	J-3	Linksys	192.168.200.3	CMP003
		Asus	192.168.200.103	CMP103
		Fonera	192.168.200.203	CMP203
4	N-3	Linksys	192.168.200.4	CMP004
		Asus	192.168.200.104	CMP104
		Fonera	192.168.200.204	CMP204
5	B-8	Linksys	192.168.200.5	CMP005
		Asus	192.168.200.105	CMP105
		Fonera	192.168.200.205	CMP205
6	F-8	Linksys	192.168.200.6	CMP006
		Asus	192.168.200.106	CMP106
		Fonera	192.168.200.206	CMP206
7	J-8	Linksys	192.168.200.7	CMP007
		Asus	192.168.200.107	CMP107
		Fonera	192.168.200.207	CMP207
8	N-8	Linksys	192.168.200.8	CMP008
		Asus	192.168.200.108	CMP108
		Fonera	192.168.200.208	CMP208
9	B-13	Linksys	192.168.200.9	CMP009
		Asus	192.168.200.109	CMP109
		Fonera	192.168.200.209	CMP209
10	F-13	Linksys	192.168.200.10	CMP010
		Asus	192.168.200.110	CMP110
		Fonera	192.168.200.210	CMP210
11	J-13	Linksys	192.168.200.11	CMP011
		Asus	192.168.200.111	CMP111
		Fonera	192.168.200.211	CMP211
12	N-13	Linksys	192.168.200.12	CMP012
		Asus	192.168.200.112	CMP112
		Fonera	192.168.200.212	CMP212
13	J-18	Linksys	192.168.200.13	CMP013
		Asus	192.168.200.113	CMP113
		Fonera	192.168.200.213	CMP213
14	N-18	Linksys	192.168.200.14	CMP014
		Asus	192.168.200.114	CMP114
		Fonera	192.168.200.214	CMP214

Tabla B.1: Disposición y direccionamiento de los equipos del proyecto

B.8 Anexo: Plano del laboratorio

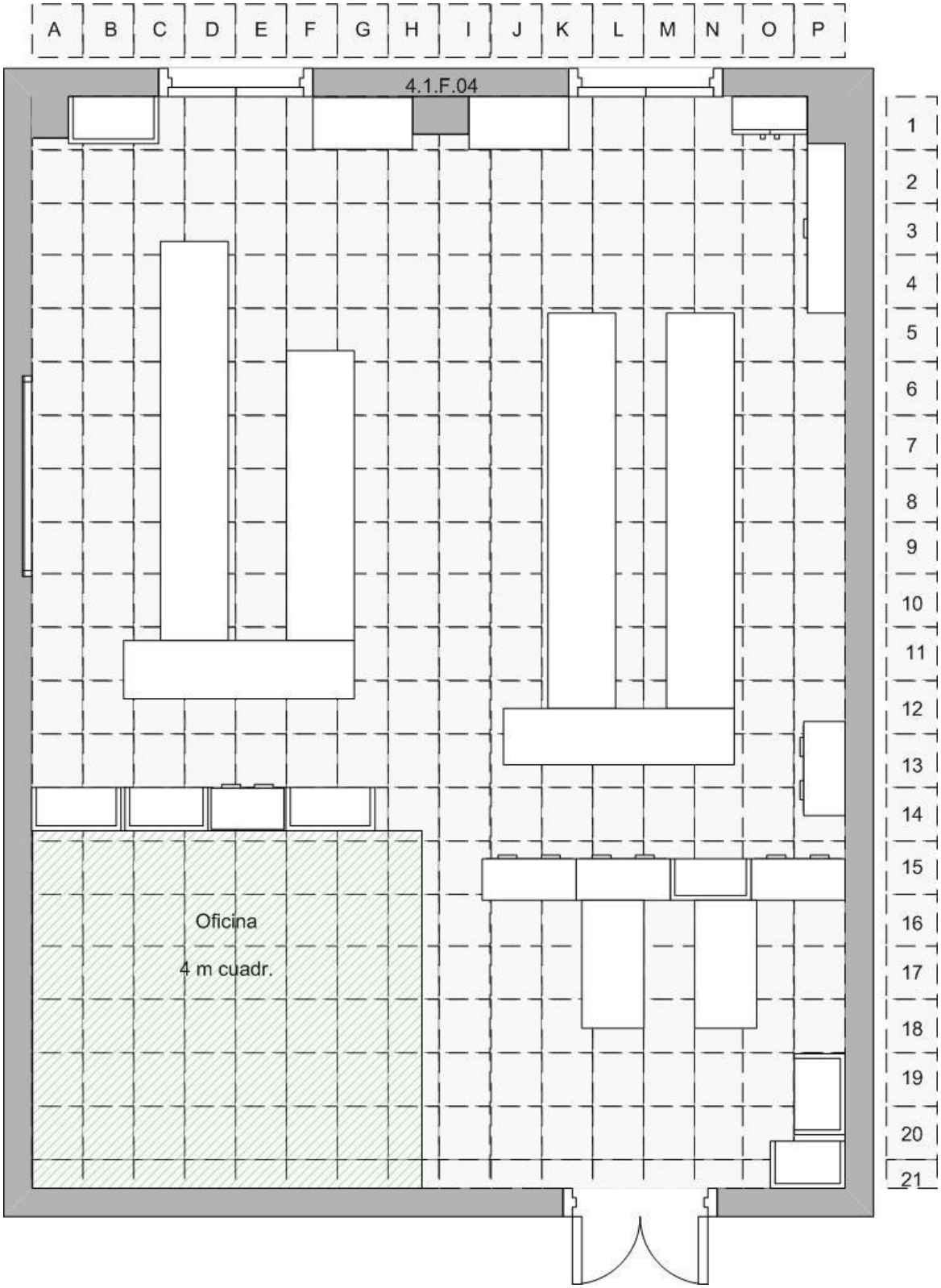


Figura B.7: Plano del Laboratorio 4.1.F04

Referencias

[1] Ian F. Akyildiz, Xudong Wang, Weilin Wang. *"Wireless mesh networks: a survey,"* Broadband and Wireless Networking (BWN) Lab, School of Electrical and Computer Engineering, Georgia Institute of Technology.

[2] Richard Draves, Jitendra Padhye, Brian Zill. *"Routing in multi-radio, multi-hop wireless mesh networks. Proceedings of the 10th annual international conference on Mobile computing and networking,"* Microsoft Research, Redmond, WA.

[3] Pablo Serrano, Antonio de la Oliva, Carlos J. Bernardos, Ignacio Soto, Albert Banchs and A. Azcorra. *"A CARMEN mesh experience: deployment and results,"* Universidad Carlos III de Madrid.

[4] Yusuke Takahashi, Yasunori Owada, Hiraku Okada, Kenichi Mase. *"A wireless mesh network testbed in rural mountain areas. Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization, "*. Niigata University, Niigata, Japan.

[5] P. Serrano, Carlos Jesus Bernardos, Antonio de la Oliva, A. Banchs, Ignacio Soto, and M. Zink *"FloorNet: Deployment and Evaluation of a Multihop Wireless 802.11 Testbed."*

[6] Javier Simó, Pablo Osuna, Joaquín Seoane, Andrés Martínez. *"Router solar autoconfigurable para redes Mesh IEEE 802.11 de telemedicina rural en América Latina,"* Fundación EHAS. Departamento de Ingeniería Telemática de la Universidad Politécnica de Madrid y

Departamento de Teoría de la Señal y Comunicaciones de la Universidad Rey Juan Carlos.

[7] Francisco Ramos Santos. *" Proyecto fin de carrera: Evaluacion de prestaciones de una red mallada basada en los dispositivos Linksys WRT54GL."*

[8] John Bicket, Daniel Aguayo, Sanjit Biswas, and Robert Morris. *"Architecture and Evaluation of an Unplanned 802.11b Mesh Network,"* Mobicom 2005, Aug 2005.

[9] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, Robert Morris. *"Link-level Measurements from an 802.11b Mesh Network,"* SIGCOMM 2004, Aug 2004.

[10] M. Bredel and M. Fidler, *"A Measurement Study of Bandwidth Estimation in IEEE 802.11 g Wireless LANs Using the DCF,"* Lecture Notes in Computer Science, vol. 4982, p. 314, 2008.

[11] C. Balanis, *"Antenna Theory and Design,"* John Wiley-Sons, Inc, pp. 249/347.

[12] RFC 1350: http://www.faqs.org/rfcs/rfc1350.html

[13] http://www.tcpdump.org/tcpdumpman.html