A mi abuelita

’Hay hombres que luchan un día y son buenos. Hay otros que luchan un año y son mejores. Hay quienes luchan muchos años, y son muy buenos. Pero hay los que luchan toda la vida, esos son los imprescindibles’

Bertolt Brecht

AGRADECIMIENTOS

En primer lugar tengo que agradecerles a mis padres su apoyo incondicional, dándome siempre la oportunidad de llegar más alto. A mi hermano, mi ejemplo y 'guía' y a Mari José, mi algo más que 'cuñá'.

A mis tutores, tanto malagueño, Salvador Luna Ramirez, como madrileño, Francisco Javier Simó. A todo el equipo de EHAS, de uno y otro lado del 'charco', y muy especialmente a Joaquín Seoane, un ejemplo como profesional y, lo que es más importante, una de las mejores personas que he conocido. Sin él esto hubiese sido muy difícil. Tampoco puedo olvidar a Larry Simmons por su ayuda en la realización del código.

Por último y, aunque sea de manera más impersonal, tengo que darle las gracias a todos mis amigos, mis compañeros 'cantores', y personas que de una forma u otra me han ayudado este año durante la realización de este proyecto.

Índice general

Índice General i

Acrónimos iv

Índice de Figuras v

Índice de Cuadros vii

1 Introducción 1

1.1 ORGANIZACIÓN DEL DOCUMENTO 2

2 Sistemas de Gestión de Red 5

2.1 OBJETIVOS Y JUSTIFICACIÓN DEL SGR 5

2.2 COMPONENTES DE UN SGR 6

2.3 PROTOCOLO SNMP 6

2.4 HERRAMIENTA DE GESTIÓN 8

3 Antiguo SGR 13

3.1 CARACTERÍSTICAS DEL ANTIGUO SGR 13

3.2 ESQUEMA GENERAL 14

3.2.1 SISTEMA GESTIONADO 15

3.2.2 SISTEMA GESTOR 16

3.3 HERRAMIENTA DE GESTIÓN: ZABBIX 16

3.4 PROBLEMAS PLANTEADOS 18

4 Nuevo SGR **21**

4.1 PARÁMETROS REQUERIDOS 22

4.1.1 PARÁMETROS IWCONFIG 22

4.1.2 PARÁMETROS DE ATHEROS en /proc/sys/dev/wifix 28

4.2 IMPLEMENTACIÓN DE LOS PARÁMETROS 30

4.2.1 FICHERO DE TEXTO: MIB 31

4.2.2 FICHERO EJECUTABLE 39

4.2.3 FUNCIONES UTILIZADAS 45

5 Pruebas **51**

5.1 ESCENARIO UTILIZADO 51

5.2 SOTFWARE NECESARIO 52

5.3 CONFIGURACIÓN EN CENTREON 54

6 Redes virtuales para gestión **57**

6.1 PPTP 58

6.2 OpenVPN 59

6.3 IPsec 60

6.4 RESULTADOS 62

6.5 CONCLUSIONES 63

7 Conclusiones y líneas futuras **65**

7.1 CONCLUSIONES 65

7.2 EVALUACIÓN DEL CUMPLIMIENTO DE LOS OBJETIVOS 66

7.3 POSIBLES MEJORAS Y LÍNEA FUTURA 67

Bibliografía **70**

A Fundación EHAS **77**

A.1 SISTEMAS DE ATENCIÓN DE SALUD PRIMARIA EN ZONAS RURALES 77

A.2 JUSTIFICACIÓN DE REDES INALÁMBRICAS EN ZONAS RURALES 78

A.3 ALTERNATIVAS TECNOLÓGICAS 80

A.3.1 WIFI 80
A.3.2 VHF 84
A.3.3 HF 84
A.3.4 WiMAX 85
A.3.5 Conexión Satelital (VSAT) 85

B La red del río Napo **87**

C Implementación de la VPN **91**
C.1 PPTP 91
C.1.1 Mikrotik: Servidor 91
C.1.2 Cliente 92
C.2 OpenVPN 93
C.2.1 Mikrotik: Servidor 93
C.2.2 Cliente 94

Acrónimos

AH	Authentication Header
CS	Centro de Salud
EHAS	Enlace Hispano-Americano de Salud
ESP	Encapsulated Security Protocol
GRE	General Routing Encapsulation
HF	High Frequency
IKE	Internet Key Exchange
IP	Internet Protocol
ISAKMP	Internet Security Association and Key Management Protocol
MIB	Management Information Base
NAT	Network Address Translation
NAV	Network Allocation Vector
NCP	Network Control Protocol
OSPF	Open Shortest Path First
PPP	Point-to-Point
PPTP	Point-to-Point Tunneling Protocol
PS	Puesto de Salud
RT	Request Tracker
SGR	Sistema de Gestión de Red
SGI	Sistema de Gestión de Incidencias
SNMP	Simple Network Management Protocol (Protocolo sencillo de gestión de Red)
SO	Sistema Operativo
SW	SoftWare
TCP	Transmission Control Protocol
TIC	Tecnologías de la Información y la Comunicación
VHF	Very High Frequency
VPN	Virtual Private Network

Índice de figuras

2.1 Esquema del protocolo SNMP 8
2.2 Situación en el árbol general del nodo Cisco 9
2.3 Esquema general de un SGR 12

3.1 Esquema del antiguo SGR 15
3.2 Paquetes utilizados en el antiguo SGR 17

4.1 Situación en el árbol general de la MIB ieee802dot11 34
4.2 Subárbol a partir del nodo dot11smt 34
4.3 Subárbol a partir del nodo dot11mac 35
4.4 Subárbol a partir del nodo dot11phy 36
4.5 Situación en el árbol general de la MIB WEHAS 37
4.6 Imagen del mib-browser 38
4.7 Ruta semántica devuelta por el comando snmptranslate 39
4.8 Ruta numérica devuelta por el comando snmptranslate 39
4.9 Esquema general del programa cuando le pide un valor el cliente al agente . 43

5.1 Esquema general del escenario de prueba 53
5.2 Resultado del comando snmpwalk para la MIB IEEE802dot11 55
5.3 Resultado del comando snmpget para la MIB IEEE802dot11 55
5.4 Resultado del comando snmpwalk para la MIB WEHAS 56
5.5 Resultado del comando snmpget para la MIB WEHAS 56
5.6 Resultado del comando snmpget para la MIB WEHAS 56

6.1 Situación inicial 57
6.2 Esquema OpenVPN 61

6.3 Esquema del escenario de prueba para VPN 62

B.1 Esquema de la Red del Río Napo (Perú) 88

Índice de cuadros

4.1 **Estándares Wifi 802.11** 24
4.2 Relación Canal-Frecuencia para la banda de 2.4GHz 25
4.3 Relación Canal-Frecuencia para la banda de 5GHz 26

Capítulo 1

Introducción

El objetivo general de este proyecto se enmarca en la mejora de las condiciones de salud de las poblaciones aisladas en países en vías de desarrollo por medio de las TIC *(Tecnologías de la Información y la Comunicación)*. Este objetivo coincide con el de la Fundación EHAS *(Enlace HispanoAmericano de Salud)* [1], promotora del proyecto y protagonista de diversos proyectos de TIC en estas zonas usando tecnologías inalámbricas.

El objetivo específico de este proyecto consiste en mejorar los sistemas de gestión de las redes inalámbricas que usan tecnologías WiFi (IEEE 802.11) [2] desplegadas por EHAS.

Para ello, en primer lugar, se ha realizado un estudio del SGR *(Sistema de Gestión de Red)* utilizado hasta entonces, analizando tanto sus puntos débiles como sus puntos fuertes, así como las causas que han llevan a su cambio, definiendo a partir de este conocimiento una nueva filosofía de SGR. Una vez hecho esto y, debido a que esta nueva filosofía está basada en el protocolo SNMP *(Simple Network Management Protocol)* [3] se ha realizado, documentado y probado una extensión del agente de gestión de red del protocolo SNMP denominado Net-SNMP [4]. Esta extensión dará soporte, por un lado, a la MIB *(Base de datos de Información de Gestión, Management Information Base, de sus siglas en inglés*) de la familia de protocolos IEEE 802.11, así como a otra MIB específica desarrollada durante el proyecto, que incluye parámetros que permiten el uso de hardware WiFi para grandes distancias, no previstos en la norma.

Aunque la extensión del agente es principalmente para las redes de EHAS, de hecho, va a procederse a su implantación a partir de octubre de este año, para rentabilizar globalmente la inversión en su desarrollo, el código y documentación quieren publicarse como software

libre, de manera que se puedan usar para cualquier fin y atraer a otros usuarios y desarrolladores que puedan mejorarlo.

1.1 ORGANIZACIÓN DEL DOCUMENTO

Este documento está organizado de la siguiente forma:

- Capítulo 1: Introducción. Es este capítulo, donde, como se ve, se hace una breve descripción de los objetivos y contexto de este proyecto.
- Capítulo 2: Sistemas de Gestión de Red. Se explica, de una manera general, en qué consiste un SGR, así como todos los factores y actores que participan en el mismo, de manera que se comprenda más fácilmente la descripción del proyecto.
- Capítulo 3: Antiguo SGR. En este capítulo se explica el SGR que se quiere sustituir, la filosofía que sigue, los motivos de esta filosofía, las causas de su cambio, así como la línea que ha seguido ese cambio.
- Capítulo 4: Nuevo SGR. En este capítulo, que es el grueso del proyecto, se explica la filosofía del nuevo SGR y, sobre todo, se expone cómo se ha desarrollado la extensión del agente de gestión del protocolo SNMP para incluir la implementación de la MIB WiFi libre.
- Capítulo 5: Pruebas. En este capítulo se expone el escenario de prueba utilizado, así como el software empleado en el mismo y su configuración. Se explican, además, las pruebas realizadas y los resultados.
- Capítulo 6: VPN (Virtual Private Network). Este capítulo, que aparentemente se sale un poco de la temática del proyecto, tiene una relación directa con él. Hay que tener en cuenta que la red gestionada y el nodo gestor no siempre pertenecerán a una misma red física. De hecho, normalmente la red gestionada será una red privada y, el nodo gestor se encontrará en un punto exterior a ella, por lo que no tendrá accesibilidad a la misma. Por lo tanto, es imprescindible poder crear una VPN entre la red gestionada y el nodo gestor si se quiere gestionar. Este capítulo se encarga del estudio de diferentes métodos para crear VPN.

- Capítulo 7: Conclusiones y líneas futuras. Este capítulo resume los objetivos alcanzados en este proyecto, así como las mejoras y complementaciones que deben hacerse al mismo para alcanzar un SGR potente y estable.

- Apéndice A: Fundación EHAS. En este apéndice se explica qué es la Fundación, el fin de la misma, el entorno general donde trabaja y, sobre todo, las tecnologías que ha ido empleando y que emplea para la comunicación en zonas rurales de países en vías de desarrollo.

- Apéndice B: Se presenta la red WiFi del río Napo, banco de trabajo de los desarrollos de EHAS y donde se implantará en primer lugar el resultado de este trabajo.

- Apéndice C: Implementación de la VPN. Se especifica cómo se ha montado y preparado el escenario de prueba para implementar la VPN.

- Apéndice D: Este apéndice se encuentra en el CD adjunto y contiene la MIB IEEE802dot11 y la MIB WEHAS.

- Apéndice E: Este apéndice se encuentra en el CD adjunto y contiene los siguientes programas implementados:

  - Implementación de la MIB IEEE802dot11
    - ieee802dot11.c
    - ieee802dot11.h
    - Makefile
    - ieee802dot11.so
  - Implementación de la MIB WEHAS
    - wfehas.c
    - wfehas.h
    - Makefile
    - wfehas.so

- Acrónimos (pág. iv): Define y explica términos y acrónimos que se usan en este proyecto.

- Bibliografía (pág. 70): Donde se completa la información aquí presentada, así como todas las fuentes utilizadas para su desarrollo.

Capítulo 2

Sistemas de Gestión de Red

A la hora de desplegar una red, hay que tener en cuenta que no sólo es importante instalarla sino también poder asegurar el correcto funcionamiento y la estabilidad de la misma a largo plazo. En este punto un SGR se convierte en algo de vital importancia ya que, gracias a él se puede tener una red monitorizada teniendo un control sobre ella, de forma que se pueda actuar ante cualquier incidencia, según la situación lo requiera, de la forma más rápida posible.

2.1 OBJETIVOS Y JUSTIFICACIÓN DEL SGR

Hay que partir del hecho de que todas las redes presentan fallos. Teniendo en cuenta esta premisa, lo que se busca es que esos fallos sean lo más evitables posible y que su resolución sea la más pronta posible también.

Una manera de evitar fallos es adelantándose a ellos. Para conseguirlo hay que utilizar una serie de herramientas que realicen un monitoreo sobre la red en funcionamiento de manera que, al observar el descontrol, cambio brusco, o evolución no deseable de algún parámetro, se puedan tomar las medidas necesarias para evitar un desastre mayor.

De igual manera, para poder solucionar un problema de manera más rápida, es conveniente, por un lado, tener un conocimiento temprano de la incidencia y, por otro, un conocimiento certero o, al menos, cercano, sobre la causa de la misma.

El conjunto que se encarga de estos cometidos es el llamado SGR.

Además, todo esto no sólo supone algo beneficioso para la red en sí, para el hecho de

que sea eficiente y eficaz, sino también sobre el factor económico. El daño de los equipos, los sistemas a reemplazar, el gasto de desplazamiento, etc., también se ven reducidos por un buen SGR. El hecho de prevenir ayuda a no tener que reparar. El hecho de reparar algo no tan grave también será menos costoso que la reparación de algo más grave. Además tener una idea de lo que hay que reparar ayuda a llevar lo necesario a los puntos donde se produce el fallo, evitando tener que realizar varios desplazamientos para, en primer lugar, averiguar cuál ha sido el fallo y en segundo lugar llevar lo necesario para repararlo.

Así mismo también ayuda a conocer el uso que los usuarios hacen de los servicios ofrecidos, comprobando su utilidad o cambiando la estrategia si no se ven útiles y necesarios.

Por todos estos factores queda clara la importancia de un buen SGR, el cual es el eje del presente proyecto fin de carrera.

2.2 COMPONENTES DE UN SGR

Para conseguir una total comprensión de un SGR, hay que tener claros los diferentes actores que participan en el mismo, así como sus funciones.

Los actores principales de un SGR son el *cliente o sistema gestor* y el *agente o elemento gestionado.*

El agente es el encargado de recolectar la información que se le pide y, en casos específicos, modificar los parámetros indicados. Mientras que el cliente, por su lado, es el que le pide al agente el valor o la modificación de los parámetros que desea y al que el agente le devuelve los valores recolectados.

Uno de los protocolos de gestión más universales y utilizados para conseguir la comunicación entre estos dos actores principales es el protocolo SNMP.

2.3 PROTOCOLO SNMP

Una primera forma de gestión consistía en mensajes ICMP Echo/Reply los cuales reportaban diferentes mensajes de error, así como información bastante importante relativa a la accesibilidad de un destino, tiempo de ida y vuelta que tardan los paquetes en alcanzar ese destino, número de hosts por lo que pasa el paquete hasta el destino y tasa de paquetes

perdidos.

No obstante, para poder complementar esta primera forma de gestión surgió el protocolo SNMP, como un protocolo de gestión simple y transitorio pero que ha tenido tanta aceptación que sigue siendo la forma de gestión universal.

El protocolo SNMP puede soportar diferentes versiones: 1, 2 y 3, que se han ido desarrollando para intentar cubrir las carencias de cada versión anterior.

La primera versión es la más simple y sus principales carencias son relativas a la seguridad. Algo que puede llegar a ser importante si tenemos en cuenta que se piden datos e, incluso, pueden modificarse, lo cual nos da un gran control sobre la red y usándolo de forma indiscriminada y por personas no autorizadas puede ser peligroso.

Sin embargo, la mejora que supuso la versión dos respecto a la primera no tiene tanto que ver con la seguridad sino por el hecho de transmitir múltiple valores. En esta segunda versión se introdujeron comandos para poder pedir valores de forma iterativa y no uno por uno.

Fue ya en la tercera versión dónde se intento solucionar el problema de la seguridad, basándose en tres tipos de servicios: autenticación, privacidad (confidencialidad) y control de acceso.

Como ya se ha comentado, este protocolo se encarga de la comunicación entre un gestor y un agente. En la Figura 2.1 que se corresponde con el esquema general del protocolo SNMP, se ve cómo desde el gestor, la acción GET se utiliza para pedir el valor de un parámetro y la acción SET para modificarlo, mientras que, desde el agente GET RESPONSE es el mensaje en el que se envía el valor del parámetro pedido y EVENT es un mensaje que puede enviar el agente de forma voluntaria, sin que le hayan pedido nada, alertando de algún anomalía en el sistema gestionado.

Para que, tanto el gestor como el agente, sepan qué parámetros utilizar y a qué se refieren estos parámetros ha de utilizarse la *MIB*, que va a ser el fichero de texto dónde se describen los parámetros a utilizar así como su posición dentro del árbol general, ya que la MIB está estructurada en forma de árbol.

Este árbol está formado por distintas ramas. En cada rama va a haber un nodo etiquetado del que, a su vez, pueden colgar otros subárboles y nodos. Cuando un nodo no tiene más subárboles ni más elementos que dependan de él hablamos de un nodo hoja.

El OID lo componen una serie de números naturales que van situando cada nodo en el

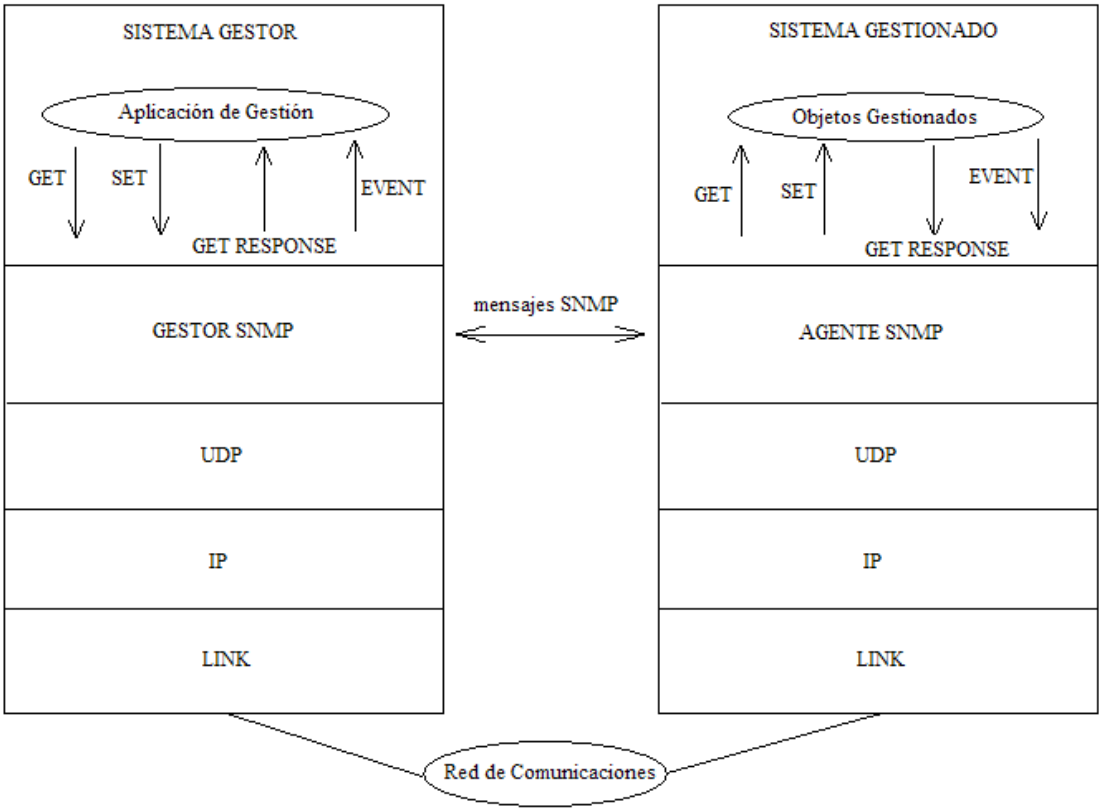

Figura 2.1: Esquema del protocolo SNMP

árbol. Así, por ejemplo, si se mira la Figura 2.2, puede verse el nodo Cisco como:

iso(1).identified organization(3).dod(6).internet(1).private(4).enterprise(1).cisco(9)

No obstante, no sólo es importante e imprescindible la definición de estos parámetros, sino la implementación de un programa que complemente al agente y que se encarga de recolectar los valores de estos parámetros.

2.4 HERRAMIENTA DE GESTIÓN

Una vez vistos lo actores principales hay que tener en cuenta otro actor muy importante: la *herramienta de gestión*, aunque un nombre más correcto sería *consola de gestión*, ya que será la interfaz con los usuarios finales.

A la hora de conseguir una buena gestión no sólo es necesario el intercambio de valores entre un agente y un cliente sino el uso que se haga de estos valores.

Gracias a las herramientas de gestión pueden utilizarse los datos que se obtienen de los

sistemas gestionados para poder analizarlos y hacer una gestión de una forma más fácil, más integral, ya que pueden incluirse todos los enlaces y dispositivos, más dinámica y con más opciones.

Dependiendo de los valores que le pida el gestor al agente y lo que posteriormente se haga con esos valores, pueden definirse diferentes formas de gestión.

De hecho, según la ISO [5], existen cinco campos de gestión de red:

Gestión de fallos. Se basa en descubrir, aislar, resolver y evitar que el problema se repita en la medida de lo posible.

Por lo tanto, la gestión de fallos incluye:

- Monitorización de la red. De manera que, con un primer vistazo a la herramienta puedan verse todos los elementos de la red y comprobar cuáles funcionan correctamente y cuáles han caído o tienen un problema.
- Disparo de alarmas. Avisando de la caída de los elementos que han tenido algún problema cuando éste se produzca.

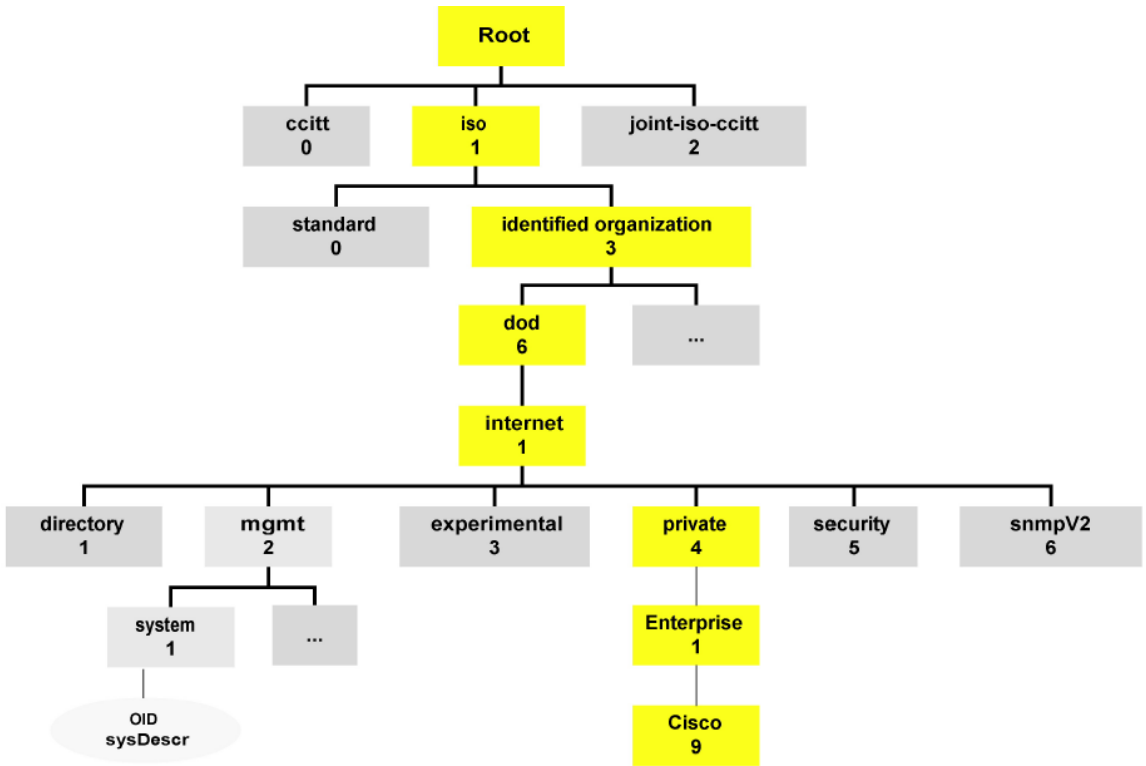

Figura 2.2: Situación en el árbol general del nodo Cisco

Estas alarmas, además, pueden configurarse de manera que sigan una jerarquía, avisando a distintas personas dependiendo del tiempo que lleve averiado un elemento, de la gravedad de la avería, de la situación de la misma, de la naturaleza de la misma, etc.

- Diagnóstico de fallos, análisis de causa raíz, solución. Analizando los valores de los parámetros de los elementos que han caído puede tenerse una primera aproximación sobre la causa de la caída, lo cual ayuda a una pronta solución del mismo.
- Etiquetado y seguimiento de problemas. Teniendo una historia de los problemas y soluciones que ha habido puede evitarse que surjan los mismos problemas o, en caso que surjan, tener constancia de la solución que se le dio contrastando si ésta fue la óptima o cambiando el protocolo de actuación.
- Gestión proactiva de los fallos. Analizando la evolución de los valores de los parámetros puede observarse si va a haber algún problema, actuando de forma que pueda evitarse.

Para complementar la gestión de fallos de la herramienta de gestión, puede utilizarse un *sistema de gestión de incidencias*. En esta herramienta se guardarán una serie de problemas, así como la fecha de cuándo surgieron, la solución que se le dio, la latencia del mismo, si volvió a repetirse, quién lo solucionó, etc. Todos los datos relevantes sobre un problema, para poder actuar de una forma más eficiente en caso de volver a repetirse. Estos problemas, además, pueden ser reportados por los mismos usuarios, no dependiendo sólo del agente y cliente, aunque también pueden detectarse de forma automática.

Gestión de configuraciones. Tiene que ver con todo lo que es la configuración de los equipos, descubrimiento de nuevos equipos, conexión con el resto de la red, detección de su configuración, almacenamiento de la misma y cambio de alguna configuración en caso en el que fuese necesario.

Gestión de seguridad. Hay que tener en cuenta que el sistema de gestión de red tiene acceso a la red, tanto para monitorizar como incluso para control y cambio de parámetros. Esto es algo muy delicado y exige que esté controlado. Por ello es necesario tener

constancia sobre qué usuarios pueden acceder a qué parte de la gestión, qué acciones realizan esos usuarios, cuándo han realizados dichas acciones, así como sus cuentas y contraseñas.

Gestión de usuarios y cuentas. Este aspecto tiene que ver más con la parte económica y facturación. Es importante saber qué usuarios acceden a qué servicios de la red, durante cuánto tiempo y en qué momentos del día. Tanto para la facturación como para el análisis estadístico de los servicios que se demandan y necesitan.

Gestión de prestaciones. Monitorización de todos los parámetros que nos interesen. Obtención de estos parámetros, graficación de los mismos, seguimiento de la evolución, relación entre unos parámetros y otros, etc.

No obstante, SNMP nace con el foco puesto en la gestión de fallos y en la gestión de configuraciones, que es dónde se hará más hincapié.

Para poder realizar este tipo de gestiones, las herramientas cuentan con una GUI (Interfaz Gráfica de Usuario), mapa de red, sistema de gestión de base de datos, métodos para interrogar a los dispositivos, menús configurables, registro de eventos y, como algo adicional, pueden generar gráficos y programación de aplicaciones API (Application Programming Interface).

Como última reseña decir que esta herramienta, que puede estar en el sistema gestor o en otro dispositivo, tiene como elemento común con el cliente una base de datos donde almacenar los valores recolectados así como una serie de demonios que, analizando esa base de datos, piden los parámetros que se necesiten.

Como esquema general, puede representarse el SGR del presente proyecto según se muestra en la Figura 2.3.

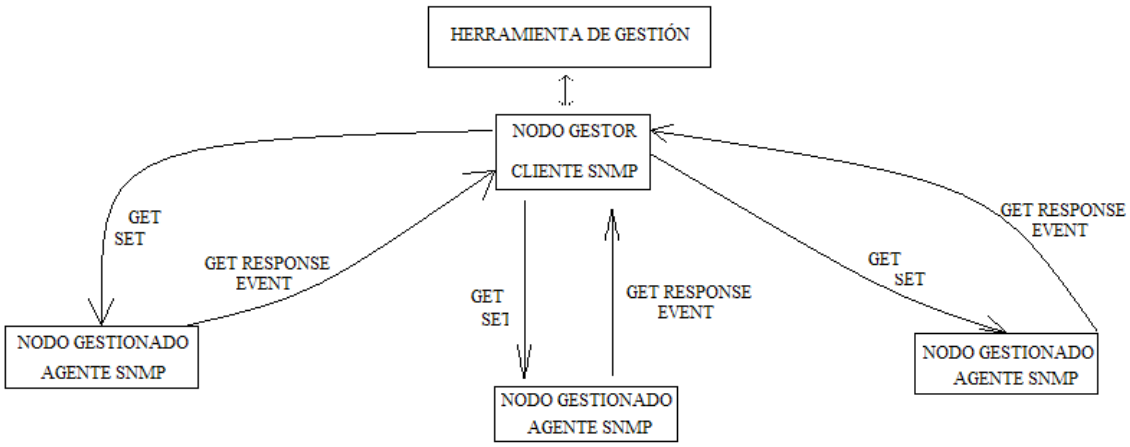

Figura 2.3: Esquema general de un SGR

Capítulo 3

Antiguo SGR

Para poder entender el SGR que se utilizaba anteriormente, hay que tener en cuenta que se gestó en un entorno y situación distintos a los que actualmente hay.

Las primeras tecnologías utilizadas para redes inalámbricas en zonas rurales eran VHF/HF. Posteriormente, se han ido utilizando otras tecnologías. De hecho WiFi ha supuesto una gran mejora para las redes inalámbricas aparcando la tecnología VHF/HF. Y actualmente se estudian otras posibilidades como WiMAX o conexiones satelitales. Se recomienda leer el Apéndice A: Fundación EHAS para entender mejor las diferentes tecnologías.

Pero, a pesar de los avances en estas tecnologías, hay que entender que el SGR que se utilizaba se había diseñado para redes con tecnologías VHF/HF. Aunque, al implantar posteriormente redes con tecnología WiFi, como es el caso de la red de estudio, la situada a lo largo del río Napo en la provincia de Loreto de Perú, se usó la misma filosofía de SGR.

Actualmente, como se ha comentado, la tecnología VHF/HF ha quedado aparcada y, por diversos motivos que ahora se expondrán, el SGR también queda obsoleto, por lo que ha sido necesario redefinir un nuevo SGR limitado a tecnologías de conexión permanente, como es la tecnología WiFi, que es la que actualmente se utiliza y donde se están centrando la mayoría de los estudios.

3.1 CARACTERÍSTICAS DEL ANTIGUO SGR

Las características del antiguo SGR hay que buscarlas en la tecnología VHF/HF para la que fue diseñado.

En principio, esta banda está pensada sólo para la transmisión de voz y, por lo tanto, los equipos radio que se utilizan se diseñan y fabrican para este fin. No obstante, mediante SW se ha conseguido utilizar este medio para comunicaciones de datos.

Gracias al paquete de SW libre *soundmodem* se consigue la comunicación entre la radio y la computadora, modulando y demodulando la señal analógica que llega de la radio a señal digital para poder utilizarla en la computadora y viceversa.

En VHF pueden alcanzarse velocidades de alrededor de 9600bps, que es poco para una óptima navegación en Internet pero, suficiente para el uso de correo electrónico.

Además, la conexión no será permanente y prima el envío de voz sobre el envío de datos. Por lo tanto y, de nuevo, el envío de información será mediante correo electrónico, ya que permite almacenar la información el tiempo necesario hasta que sea posible tener conexión.

Esto supuso que el SGR estuviese basado en el envío de *correo electrónico* entre el equipo que se quería gestionar y el equipo gestor, en el cuál iba la información necesaria para poder realizar la gestión.

Este intercambio de información se realiza, por lo tanto, mediante el protocolo SMTP o BSMTP sobre UUCP sobre TCP, protocolos del nivel de aplicación para el envío de correo electrónico.

Puede observarse que no son protocolos típicos de gestión de red, ya que estos no podían utilizarse en redes VHF/HF. Sin embargo, al fin y al cabo, para poder gestionar lo que se necesitan son los valores de los parámetros que se quieran monitorizar. El antiguo SGR lo que hacía era recolectar estos valores de los sistemas gestionados y enviarlos por correo electrónico al sistema gestor para que éste los utilizase.

3.2 ESQUEMA GENERAL

El esquema general del antiguo SGR está formado por los siguientes bloques que se ven en la Figura 3.1

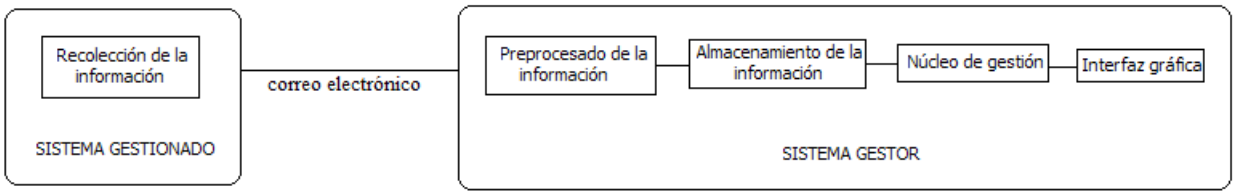

Figura 3.1: Esquema del antiguo SGR

3.2.1 SISTEMA GESTIONADO

En los sistemas gestionados, que se corresponden con los ordenadores y los routers, lo que hay que hacer es recolectar la información necesaria que quiere ser monitorizada y gestionada, empaquetarla y enviarla como adjunto de un correo electrónico al sistema gestor.

La forma de obtener la información es mediante la ejecución periódica de unos *script*, programas que ejecutarán una serie de comandos que devolverán los valores requeridos, así como la consulta de los logs, unos registros de eventos que se guardan durante un periodo de tiempo. En ellos se almacenan los datos así como la información sobre quién, qué, cuándo, dónde y por qué un evento concreto ocurrió.

La información que va a recolectarse puede dividirse en varios grupos.

- Información del sistema. Se envía cada 15 días porque es información estática como por ejemplo, el nombre del equipo, paquetes instalados, tablas de particiones, sistema de archivos, buses, impresoras conectadas al equipo, información de configuración del equipo.

- Información a corto plazo. Se corresponde con los datos que van a ir recolectándose de forma periódica porque varían continuamente como por ejemplo, la temperatura y estado de la CPU, información general de las interfaces como dirección IP, dirección MAC, paquetes, bytes y errores de transmisión y recepción, para las interfaces inalámbricas además, el punto de acceso y frecuencia, velocidad de transmisión y recepción, SNR.

- 'Alive'. Es un mensaje que se envía cada cierto tiempo si existe actividad en el dispositivo, el hecho de que no se envíe indica que el dispositivo no está activo. El tiempo de envío puede variar entre 2-8 horas, 2-6 ó 15 días.

- Información diaria. Normalmente se corresponde con información estadística como el tiempo que ha estado encendido el dispositivo, número de correos entrantes y salientes, número de llamadas realizadas.

3.2.2 SISTEMA GESTOR

Una vez que al sistema gestor le ha llegado mediante correo electrónico la información que necesita, tiene que organizarla, almacenarla y utilizarla como corresponda.

Para ello cuenta con una serie de módulos.

En primer lugar, el módulo de *preprocesado de la información*, tiene que separar el fichero adjunto del correo que llega. A continuación tiene que almacenar una copia de seguridad de ese adjunto en la carpeta que se corresponda con el sistema gestionado del cual procede, tras lo cuál se extrae la información importante.

El módulo de *almacenamiento de la información*, cogerá esa información importante y la almacenará en la base de datos de la herramienta de gestión que vaya a utilizarse, especificando el momento en el que se obtuvieron esos datos.

El módulo de *núcleo de gestión* es el que se corresponde con la herramienta de gestión en sí, es decir, es la herramienta que utilizará los datos almacenados para monitorizarlos y realizar las acciones que se vayan a programar, para hacerlo de una forma más fácil e intuitiva se ayudará del módulo de *interfaz gráfica*.

3.3 HERRAMIENTA DE GESTIÓN: ZABBIX

Para poder realizar la monitorización y seguimiento de la red se utilizó como herramienta de gestión Zabbix, una herramienta de SW libre, la cuál hubo que adaptar para que pudiese utilizarse según el esquema anterior.

Concretamente, además de los módulos propios de Zabbix, se tuvieron que incluir, dentro del gestor, los módulos *Control SGRE (Sistema de Gestión de Red EHAS), Modelo SGRE y Vista SGRE*, como puede verse en la Figura 3.2

Concretamente *Control SGRE* se corresponde con las acciones de recepción de correo, procesamiento de ese correo, adición de equipos en el caso en el que lleguen correos de

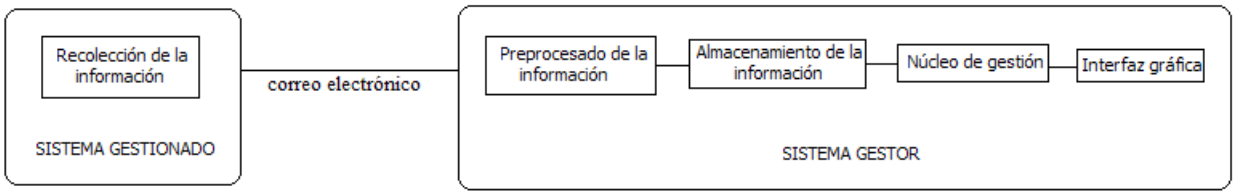

Figura 3.2: Paquetes utilizados en el antiguo SGR

equipos no registrados y procesamiento de la información recolectada en dichos correos.

Básicamente lo que hace Zabbix es el almacenamiento en la base de datos de la información que le llegue. Por lo tanto, este es el elemento que más hay que modificar ya que el resto de la acciones (recepción de correo, procesamiento de correo, adición de equipos y obtención de la información necesaria) han tenido que ser implementadas y añadidas a Zabbix para que funcione como se busca.

Así pues se ha añadido un decodificador MIME, en concreto *munpack*, que se encarga de extraer los archivos adjuntos desde el mismo archivo de correo. Además de la herramienta Procmail, que es un procesador de mensajes de correo electrónico muy versátil, se utiliza para clasificar los mensajes de correo en función de diferentes criterios y una serie de script que se encargan de obtener la información necesaria.

Las componentes modelo y vista no han tenido que ser tan modificados.

La componente *modelo* se centra en la manipulación de la información almacenada en la base de datos y Zabbix cumplía con los requisitos buscados, encargándose de las siguientes tareas:

- Gestión de alarmas. Detecta averías u otros problemas, intentando repararlos o enviando la alarma a quién corresponda.
- Gestión de inventario. Se encarga de mantener la información inventariable del equipo (hardware, software instalado,...)
- Configuración. Modificación en los sistemas gestionados los parámetros que se crean importantes.
- Correlación de eventos. Esto es especialmente importante para evitar el 'efecto tomate', que surge cuando por una única causa se disparan un número de alarmas en serie.

Respecto a la componente *vista*, la cual se encarga de hacer más agradable el acceso a la información gestionada por parte de los usuarios, decir que se ha heredado prácticamente de Zabbix, aunque se han incluido algunas modificaciones en gráficas, tablas, mapas, listado de alarmas y configuración.

3.4 PROBLEMAS PLANTEADOS

En realidad, el sistema diseñado cumplía con el propósito que buscaba, consiguiendo una gestión bastante completa.

El problema principal viene derivado de la mantenibilidad de la herramienta de gestión, sobre todo.

Al analizar el sistema se ha visto cómo, para que la herramienta de gestión fuese compatible con un sistema basado en correo electrónico, se ha tenido que modificar internamente, instalando una serie de script y paquetes adicionales para poder recoger estos correos y extraer su información.

Esto implica que, al actualizarse Zabbix todo lo añadido a la herramienta sea incompatible con la actualización. Provocando que no pueda gestionarse a no ser que se adaptase lo añadido a la nueva actualización.

Sin embargo, las personas encargadas de la mantenibilidad del sistema, las cuales no tenían por qué ser las que lo habían implementado, lo encontraban complejo y muchas veces incomprensible y, por lo tanto, difícil de mantener.

Otro problema, derivado de los usuarios finales, se encontraba en la interfaz gráfica. Consideraban que tenía demasiada información, que no siempre era fácil de usar y era bastante lenta, por lo que no llegaron a empatizar con ella.

Por último quedaba pendiente el hecho de poder gestionar sistemas cerrados a los que no se tuviese acceso. Ya que no se utiliza ningún protocolo de gestión sino que se pide la información y se envía por correo electrónico, se tiene que tener acceso al sistema y no se puede utilizar de forma universal.

Todo esto llevó al planteamiento de un nuevo SGR alternativo. Ya no sólo en su manera de actuación, buscando una forma más universal que la basada en correo electrónico, sino también en la forma de llegar a los usuarios finales, es decir, en la herramienta de gestión.

3.4. PROBLEMAS PLANTEADOS

El protocolo de gestión más universal y, por lo tanto, mantenible y disponible en la mayoría de los sistemas, es decir, el protocolo SNMP, fue la opción tomada.

Esta decisión fue posible porque en la actualidad se opta por las redes WiFi, con mayor ancho de banda, en las que se puede enviar todo tipo de datos, primera restricción que imponían las redes VHF/HF. Además cuentan con conectividad permanente, segunda restricción que imponían las redes VHF/HF, por lo que se puede acceder al sistema gestionado y preguntarle por la información necesaria cuando se necesite.

Por lo tanto, se ha encaminado el SGR por el *protocolo SNMP* y se ha buscado una nueva herramienta de gestión para poder tratar los valores recolectados. Esta herramienta ha sido *Centreon*.

Al elegir la opción del protocolo SNMP, el principal problema lo plantearon los routers.

Cuando se instala un agente de gestión SNMP, Net-SNMP, en los distintos sistemas, por defecto, pueden gestionarse una serie de parámetros bastantes útiles. No obstante, los parámetros WiFi que interesaban no podían ser gestionados ya que no existía la implementación libre de los mismos. Por lo tanto, el fin principal de este proyecto fin de carrera ha sido la extensión del agente Net-SNMP, definiendo e implementando los parámetros WiFi que se necesitaban gestionar.

Capítulo 4

Nuevo SGR

Como se comentó en el capítulo anterior, la base del nuevo SGR se centra en el protocolo SNMP.

Dentro del protocolo, el principal objetivo de este proyecto va a centrarse en el *agente*, es decir, en la entidad encargada de obtener los valores de los parámetros que le pida el cliente. El agente libre que se va a utilizar para el protocolo SNMP en Linux [15] [1] es el llamado Net-SNMP [2].

Como también se mencionó, por defecto, este agente es capaz de recolectar una serie de valores de unos parámetros concretos. No obstante, como para el caso de los parámetros WiFi, parámetros imprescindibles de gestionar en la red de estudio, el agente no tiene definidos ni los parámetros ni la forma de obtener estos valores. Pero, gracias a que el agente es extensible, en este proyecto, por un lado, se han definido los parámetros que interesaban ser gestionados, es decir, la MIB, así como su implementación, es decir, la forma en la que el agente va a obtener los valores de esos parámetros. Una vez hecho esto, los parámetros WiFi de los routers podrán ser gestionados mediante SNMP con SW libre, cosa que no podía hacerse hasta ahora.

[1] Durante todo el proyecto el SO utilizado ha sido Ubuntu Hardy 8.04 basado en Linux, ya que la filosofía de la Fundación EHAS es el uso de SW libre

[2] Se instala el paquete con el agente, sus ficheros de configuración y MIBs por defecto y se accede al mismo mediante línea de comandos

4.1 PARÁMETROS REQUERIDOS

A la hora de comenzar la implementación de cualquier MIB lo primero que hay que tener claro es qué parámetros van a ser los requeridos, los cuales vamos a monitorizar y, por lo tanto, hay que implementar.

En este proyecto los parámetros pedidos van a ser los obtenidos al ejecutar el comando *iwconfig* en un sistema Linux y los parámetros que aparecen en */proc/sys/dev/wifix* en un sistema Linux con tarjeta inalámbrica con chipset Atheros[16], las utilizadas en la red de actuación.

4.1.1 PARÁMETROS IWCONFIG

En primer lugar se van a explicar todos los parámetros que aparecen el ejecutar el comando *iwconfig:*

```
IEEE 802.11g  ESSID:"WLAN_3C"
Mode:Managed  Frequency:2.422 GHz  Access Point: 00:1A:2B:06:1D:1F
Bit Rate:54 Mb/s   Tx-Power=20 dBm   Sensitivity=8/0
Retry limit:7   RTS thr:off   Fragment thr:off
Power Management:off
Link Quality=93/100  Signal level=-34 dBm  Noise level=-88 dBm
Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

1. *Estándar Wifi utilizado*. Este parámetro va a especificar el estándar 802.11 usado. Entre otros aspectos el estándar me informa sobre la frecuencia de funcionamiento, codificación y velocidad binaria utilizada. En la Tabla 4.1 puede observarse un resumen de los estándares 802.11 aunque, en realidad, los que se utilizan en nuestro caso de estudio son el 802.11g, 802.11b y 802.11a.

4.1. PARÁMETROS REQUERIDOS

IEEE 802.11	Estándar para LANs inalámbricas (WLANs) en la Banda de Frecuencias de 2400 MHz a 2484 MHz. Se diferencia del IEEE 802.3 en las Capas Física y de Enlace de Datos. Utiliza espectro ensanchado por secuencia directa (DSSS) y tiene tasas de transferencia de 1 y 2 Mbps. Puesto que estas 2 velocidades de transmisión se encuentran generalmente presentes en la interfaz IEEE 802.11b, es que se las conoce también como IEEE 802.11b, o directamente 802.11b (a pesar de diferir en sus especificaciones).
IEEE 802.11a	Estándar para LANs inalámbricas (WLANs) en las Bandas de Frecuencias de 5,150 GHz a 5,350 GHz y de 5,725 GHz a 5,850 GHz. Utiliza multiplexación por división de frecuencias ortogonales (OFDM), alcanzando tasas de transferencia 6, 9, 12, 18, 24, 36, 48 y 54 Mbps, siendo las obligatorias a implementar 6, 12 y 24 Mbps.
IEEE 802.11b	Estándar para LANs inalámbricas (WLANs) en la Banda de Frecuencias de 2400 MHz a 2484 MHz. Utiliza espectro ensanchado por secuencia directa de alta tasa (HR/DSSS), siendo estas de 5,5 y 11 Mbps.
IEEE 802.11g	Estándar para LANs inalámbricas (WLANs) en la Banda de Frecuencias de 2400 MHz a 2484 MHz. Utiliza la misma tecnología OFDM que IEEE 802.11a, alcanzando tasas de transferencia 6, 9, 12, 18, 24, 36, 48 y 54 Mbps, pero en las mismas frecuencias de IEEE 802.11b.
IEEE 802.11h	Estándar para LANs inalámbricas (WLANs) en las Bandas de Frecuencias de 5,150 GHz a 5,350 GHz y de 5,725 GHz a 5,850 GHz. Variante europea de IEEE 802.11a, pero con la capacidad de seleccionar dinámicamente la frecuencia (DFS) y controlar la potencia de transmisión (TPC) para resolver los problemas derivados de la coexistencia con sistemas de Radares y Satélites en estas bandas, utilizadas generalmente por sistemas militares.
IEEE 802.11j	Estándar para LANs inalámbricas (WLANs) en las Bandas de Frecuencias de 5,150 GHz a 5,350 GHz y de 5,725 GHz a 5,850 GHz. Variante japonesa de IEEE 802.11a, pero con la capacidad de seleccionar dinámicamente la frecuencia (DFS) y controlar la potencia de transmisión (TPC) para resolver los problemas derivados de la coexistencia con sistemas de Radares y Satélites en estas bandas, utilizadas generalmente por sistemas militares.
IEEE 802.11n	Estándar para LANs inalámbricas (WLANs) en las Bandas de Frecuencias de 2400 MHz a 2484 MHz, de 5,150 GHz a 5,350 GHz y de 5,725 GHz a 5,850 GHz. Utiliza la misma tecnología OFDM que IEEE 802.11a e IEEE 802.11g, más la tecnología Múltiples Entradas - Múltiples Salidas (MIMO), alcanzando tasas de transferencia de hasta 600 Mbps.

IEEE 802.11e	Añade al estándar 802.11 factores como calidad de servicio (QoS), permitiendo soportar tráfico en tiempo real en todo tipo de entornos y situaciones.
IEEE 802.11i	Intenta resolver los elementos de seguridad y encriptación del estándar básico. El estándar abarca los protocolos 802.1x, TKIP y AES. Se implementa WPA2.

Cuadro 4.1: **Estándares Wifi 802.11**

2. *ESSID*. Este parámetro representa a un conjunto de células que pertenecen a la misma red virtual. Esto quiere decir que, a lo largo de estas células y, gracias a los repetidores e infraestructuras de las mismas, el cliente puede viajar (roaming) de forma transparente. Todos los dispositivos que intenten comunicarse entre sí deben de tener el mismo ESSID.

3. *Mode*. Este parámetro representa el modo de funcionamiento del dispositivo o nodo concreto. Estos modos pueden ser:

   - Ad-Hoc. En este modo la red está compuesta por una sola célula y no existe punto de acceso.
   - Managed. En este modo el nodo se va conectando a la red que está compuesta por varios puntos de acceso. El nodo puede ir viajando por la red cambiando el punto de acceso.
   - Master. En este modo el nodo es el responsable de la sincronización o bien actúa como si él fuese el punto de acceso.
   - Repeater. En este modo el nodo lo único que hace es reenviar paquetes a otros nodos wireless.
   - Secondary. En este modo el nodo actúa como un respaldo del master.
   - Monitor. En este modo el nodo no se asocia a ninguna célula en concreto y, de forma pasiva, monitoriza todos los paquetes de la frecuencia.

4. *Frequency*. Este parámetro especifica la frecuencia o canal de funcionamiento concreto del nodo. Existe una lista de canales y frecuencias disponibles. Dependiendo del reglamento puede que algunas frecuencias/canales no estén disponibles. En las Tabla

IDENTIFICADOR DE CANAL	FRECUENCIA (MHz)
1	2412
2	2417
3	2422
4	2427
5	2432
6	2437
7	2442
8	2447
9	2452
10	2457
11	2462
12	2467
13	2472
14	2484

Cuadro 4.2: Relación Canal-Frecuencia para la banda de 2.4GHz

4.2 y Tabla 4.3 se muestra la relación frecuencia-canal tanto para la banda de 2.4GHz como la de 5GHz [17].

5. *Access Point*. Este parámetro me especifica la dirección MAC del punto de acceso, el cual es el dispositivo inalámbrico central de una red Wifi. A él y, por medio de ondas de radio frecuencia, llegan los paquetes de los diferentes dispositivos móviles y él se encarga de transmitirlos a donde corresponda, bien sea otro dispositivo inalámbrico bien sea a una red cableada.

6. *Bit Rate*. Este parámetro va a especificar la velocidad a la cual los bits van a ser transmitidos por el medio. Aunque, en realidad, la velocidad de usuario será menor debido a diferentes factores como obstáculos, el hecho de que el medio es compartido o el hecho de que no todos los paquetes enviados representen informacin útil.

7. *Tx-Power*. Representa la potencia transmitida en dBm.

8. *Sens*. Representa la sensibilidad del nodo. Este parámetro puede referirse a diferentes aspectos. Por un lado, la sensibilidad representa el mínimo valor de señal para el cuál el nodo sigue asociado a un punto de acceso específico. Por debajo de ese nivel el nodo

IDENTIFICADOR DE CANAL	FRECUENCIA (MHz)
34	5170
36	5180
38	5190
40	5200
42	5210
44	5220
46	5230
48	5240
52	5260
56	5280
60	5300
64	5320
149	5745
153	5765
157	5785
161	5805

Cuadro 4.3: Relación Canal-Frecuencia para la banda de 5GHz

comienza a buscar otro punto de acceso al cual asociarse. Normalmente, si existe un número grande de puntos de acceso es bueno que este valor sea mayor para asegurar así la asociación al mejor punto de acceso. Si hay un número bajo de puntos de acceso conviene que este nivel sea menor para evitar el constante intento de cambio de punto de acceso. Otra forma de ver este parámetro es considerando que es el menor nivel de señal para transmitir. De manera que si no se detecta un nivel de señal mayor, se considera el canal libre para transmitir. También sirve como umbral entre la señal que se considera válida y el ruido.

9. *Retry.* Este parámetro representa el tiempo máximo, expresado en segundos por defecto, en el que la capa MAC va a retransmitir el paquete antes de descartarlo. Dependiendo si se aplica el protocolo RTS/CTS o no, el tiempo de retransmisiones es mayor o menor, respectivamente.

10. *RTS thr.*Este parámetro define el tamaño mínimo de paquete para el que va a ejecutarse el protocolo RTS/CTS. Este protocolo asegura que el canal está libre antes de

transmitir. Para asegurarlo el AP envía un paquete listo para enviar a la estación que quiera hacerlo de manera que no haya colisiones a la hora de transmitir. Para tamaños menores a ese mínimo no habrá que ejecutar el protocolo. Por lo tanto, teniendo este valor igual al tamaño máximo de paquete se deshabilita el protocolo.

11. *Fragment thr.*Hay ocasiones en las que los paquetes IP superan el tamaño máximo de transmisión. En estos casos es necesario fragmentar los paquetes para poder enviarlos. Este parámetro va a especificar el tamaño máximo de fragmento en los que se podrá dividir el paquete IP y que, evidentemente, siempre será menor al tamaño máximo de paquete de transmisión. Por lo tanto, teniendo este valor igual al tamaño máximo de paquete se deshabilita el mecanismo de fragmentación.

12. *Power Management.*Este parámetro da la oportunidad de controlar varios parámetros como por ejemplo, período, expresado en segundos por defecto, en el que el nodo permanece dormido, es decir, en modo ahorro de energía, período, expresado en segundos por defecto, en el que el nodo permanece despierto, es decir, en modo activo, hasta volver al estado de dormido. Además, puede expresar el tipo de paquetes que debe recibir el nodo: todos, sólo unicast, descartando tanto multicast como broadcast, o multicast, recibiendo paquetes tanto multicast como broadcast y descartando paquetes unicast.

13. *Link Quality.* Este parámetro representa la calidad del enlace en tanto por ciento. El cálculo de este parámetro se basa en el nivel de interferencia, BER, tasa de error de paquete, nivel de señal recibida y otros cálculos que realiza el hardware. Por lo tanto, y dado que es un conjunto de todos estos factores, este parámetro depende del driver y del hardware y no tanto del enlace en sí.

14. *Signal level.* Este parámetro representa el nivel de señal recibido en dBm. En el caso de encontrarse en modo Ad-hoc puede ser que este parámetro no esté definido.

15. *Noise level.* Este parámetro representa el nivel de ruido existente en el nodo, expresado en dBm, cuando ningún paquete está siendo transmitido.

16. *Rx invalid nwid.* Este parámetro representa el número de paquetes recibidos con ESSID o NWID (network id) diferentes. Gracias a este parámetro pueden detectarse proble-

mas de configuración a la hora de intentar formar parte de una red. También puede servir para detectar redes adyacentes existentes.

17. *Rx invalid crypt.* Este parámetro representa el número de paquetes que no ha sido posible descifrar. Por lo tanto sirve para detectar posibles problemas con el cifrado.

18. *Rx invalid frag.* Este parámetro representa el número de paquetes que no se pudieron reconstruir porque los fragmentos de los que estaba formado no eran correctos o porque se perdieron.

19. *Tx excessive retries.* Este parámetro representa el número de paquetes que, por haber superado el número de retransmisiones sin éxito, fueron descartados.

20. *Invalid misc.* Este parámetro representa el número de paquetes que se perdieron debido a cualquier otro motivo no descrito anteriormente.

21. *Missed beacon.* Los paquetes beacon o baliza son las tramas enviadas periódicamente por un AP para difundir su presencia, así como la información necesaria para poder identificar la red a la que pertenece. Este parámetro representa la pérdida de estos paquetes. El hecho de que se pierdan puede indicar que la tarjeta se está saliendo de rango.

4.1.2 PARÁMETROS DE ATHEROS en /proc/sys/dev/wifix

A continuacin se van a explicar todos los parámetros que aparecen en */proc/sys/dev/wifix*. Estos parámetros van a representar características propias de las chipset atheros.

1. *ACKRATE.* Este parámetro representa la velocidad a la que van a ser enviados los ack. En caso en el que este parámetro esté a 1 los ack son enviados a una velocidad elevada, mientras que, en el caso en el que esté a 0 los ack se envían a una velocidad normal.

2. *ACKTIMEOUT.* Este parámetro representa el tiempo máximo, expresada en microsegundos, en el que se espera la recepción de un ack (paquete que indica la correcta recepción de otro paquete enviado) tras enviar una trama de datos. Pasado este tiempo se considera que la trama no ha llegado al receptor.

3. *COUNTRYCODE.* Este parámetro representa el valor entero que se le asigna a cada país para saber qué frecuencias y canales pueden ser utilizadas en el mismo ya que, como se dijo anteriormente, no todas están permitidas.

4. *CTSTIMEOUT.* Este parámetro representa el tiempo máximo, expresado en microsegundos, en el que se espera la recepción de una trama CTS. Pasado este tiempo se considera que el medio está ocupado por lo que no se procede al envío de los paquetes.

5. *DEBUG*. Este parámetro representa a un booleano en el que 1 expresa que la opción de depuración está habilitada mientras que 0 expresara lo contrario.

6. *DIVERSITY.* La diversidad es un método mediante el cual se usan varias antenas para la recepción de la señal. De manera que la señal que finalmente se tiene es una combinación de las mismas. De esta forma se compensan las interferencias multiruta existentes y se mejora la conectividad wireless. Este parámetro representa a un booleano en el que 1 expresa que la diversidad está habilitada mientras que 0 expresará lo contrario.

7. *FFTXQMIN.* Este parámetro representa el caudal mínimo de transmisión mediante la técnica Fast Frame. Esta técnica sirve para elevar la tasa de transferencia (throughput). Para ello aumenta el tamaño de la trama (puede aumentarlo hasta 3000 bytes) y elimina pausas entre tramas, variando el ritmo de transferencia (timing).

8. *INTMIT.* Este parámetro indica si están habilitadas o no las técnicas encargadas de reducir las interferencias.

9. *LEDPIN.* Este parámetro representa el valor que se refiere al led que indica la existencia de conexión WiFi. Este valor es distinto en cada laptop y puede ser cualquier natural.

10. *REGDOMAIN.* Al igual que para *countrycode*, este parámetro representa un valor entero que se le asigna a una región, en la que hay incluidos varios países, para saber qué frecuencias y canales pueden ser utilizadas en la misma. Por lo tanto, *regdomain* es una región mayor a la de *countrycode*, que es más especfica.

11. *RXANTENNA*. Este parámetro representa cuál de las antenas del dispositivo es la que es receptora. 0 indica que la antena principal es la receptora mientras que 1 indica que la receptora es la antena secundaria.

12. *SLOTTIME.* Este parámetro, expresado en microsegundos, representa el número de intervalos en que se divide la ventana de contienda.

13. *SOFTLED.* Valor complementario a *ledpin* que indica el led que se enciende para representar que existe una conexión inalámbrica.

14. *TKIPMIC.* Este parámetro es un booleano en el que 1 indica que se utiliza el protocolo TKIP (Temporary Key Interval Protocol), gracias al cual la clave de la conexión varía transcurrido un período de tiempo específico, mientras que 0 indica la no utilización del mismo.

15. *TXANTENNA.* Análogo al parámetro RXANTENNA (11), este parámetro representa cuál de las antenas del dispositivo es la que es transmisora. 0 indica que la antena principal es la transmisora mientras que 1 indica que la transmisora es la antena secundaria.

16. *TXINTERPERIOD.* Este parámetro representa el tiempo, expresado en segundos, que transcurre entre transmisiones.

17. *XRPOLLCOUNT*. Este parámetro y el siguiente tienen que ver con el chipset atheros XR, es decir, eXtended Range.

18. *XRPOLLPERIOD*

4.2 IMPLEMENTACIÓN DE LOS PARÁMETROS

Una vez que se tienen claros los parámetros que van a utilizarse se pasa a la implementación de los mismos para extender el agente.

Esta implementación va a constar de dos pasos principalmente, como se viene comentando. Por un lado, se tiene que realizar el *fichero de texto*, *MIB*, en el cuál se describe cada uno de los parámetros así como su situación dentro del árbol general. Y, en segundo lugar, hay

que realizar un *fichero ejecutable dinámico* donde se programará la forma de obtener esos parámetros y el enlace del mismo con el agente.

4.2.1 FICHERO DE TEXTO: MIB

Estructura de una MIB

Para escribir un fichero MIB va a utilizarse el lenguaje SMI que es un subconjunto de ASN.1 [18].

La estructura general del fichero será la siguiente:

- En primer lugar debe especificarse el nombre de la MIB así como la palabra clave *BEGIN*. Así mismo deben referenciarse los tipos de datos que están definidos en otras MIB y van a utilizarse. Esto se hace *importando* los tipos de datos que se necesiten *desde* las MIB que se especifiquen. Hay que destacar que todos los ejemplos que se exponen a continuación se corresponden con la MIB Wifi EHAS que se ha creado.

```
WEHAS-MIB DEFINITIONS ::= BEGIN
    IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
       Integer32, Counter32, Unsigned32
    FROM SNMPv2-SMI
       DisplayString , MacAddress, RowStatus,
       TruthValue
    FROM SNMPv2-TC
       MODULE-COMPLIANCE, OBJECT-GROUP,
       NOTIFICATION-GROUP
    FROM SNMPv2-CONF
       ifIndex
    FROM RFC1213-MIB;
```

- En segundo lugar debe definirse la posición que el nodo principal de la MIB va a tomar dentro de la estructura del árbol.

```
-- ************************************************************************
--   Tree Definition
-- ************************************************************************

    identi-orga     OBJECT IDENTIFIER ::= { iso 3 }
    dod             OBJECT IDENTIFIER ::= { identi-orga 6 }
    internet        OBJECT IDENTIFIER ::= { dod 1 }
    experimental    OBJECT IDENTIFIER ::= { internet 3 }
```

- A continuación debe ponerse una descripción general de lo que va a contener la MIB y que tiene que estar compuesta por los siguientes campos como mínimo: Last-Updated, en la que se especifica la última fecha de modificación, Organitation y Contact-Info, en la que se da el mail de contacto de la persona que ha escrito la MIB así como la organización para la que lo ha hecho, y una breve descripción sobre qué parámetros va a contener la MIB, así como su situación dentro del árbol general.

```
-- ************************************************************************
--   MODULE IDENTITY
-- ************************************************************************
wfehas  MODULE-IDENTITY
          LAST-UPDATED "0906250000Z"
          ORGANIZATION "Fundación EHAS, Madrid, Spain"
          CONTACT-INFO
           "e-mail: ana.pastor@ehas.org"
          DESCRIPTION
          "The MIB module for wifi entities handled through
           the MadWifi driver.
 iso(1).identi-orga(3).dod(6).internet(1).experimental(3).wfehas(6789)"
    ::= { experimental 6789 }
```

- A partir de aquí viene la descripción de los distintos bloques y parámetros. Los parámetros tienen que estar descritos por una serie de campos obligatorios, que se corresponden con el nombre del objeto, el tipo de dato, el tipo de acceso, esto es, si es lectura, escritura o ambos, el estado, una breve descripción y su situación respecto a su nodo padre.

```
Name OBJECT-TYPE
        SYNTAX OCTET STRING(SIZE(0..32))
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "This attribute shall indicate the wireless mode
             that is being used by the station.
             Between the options it can be found
            802.11a, 802.11b or 802.11g"
    ::= { iwconparamEntry 1 }
```

IEEE802dot11-MIB

A la hora de escribir la MIB, resultó de gran interés el hecho de que el IEEE tuviese una MIB estándar escrita para 802.11. Partiendo del estudio de ésta y para mantener siempre la mayor estandarización posible, se buscaron los parámetros que se requerían y que ya estuviesen descritos en esta MIB, dejando para otra MIB adicional, que habría que escribir, los parámetros que siendo necesarios para el proyecto, no estuviesen incluidos en la MIB estándar.

Esta MIB se encuentra situada en la siguiente posición dentro del árbol estándar:

iso(1).member-body(2).us(840).ieee802dot11(10036)

Como puede observarse en la Figura 4.1, está subdividida en cuatro campos principales.

dot11smt(Station ManagemenT) Estos parámetros representan todos los factores que son necesarios y que hay que tener en cuenta para que una estación pueda formar parte de una red WiFi. Ejemplos de estos parámetros son la direccin MAC del AP, el SSID deseado y todos los parámetros que tienen que ver con una conexión segura, tanto claves, como algoritmos, protocolos de seguridad, etc.

En la Figura 4.2, pueden verse los nodos que cuelgan de este nodo principal y que se corresponden con *tablas* en las que se incluyen los diferentes parámetros.

De todos estos parámetros los que van a interesar para el presente proyecto son básicamente la dirección *MAC del AP* y la *ESSID* deseada, ambos son hojas que cuelgan de dot11smt 1, es decir, dot11StationConfigTable.

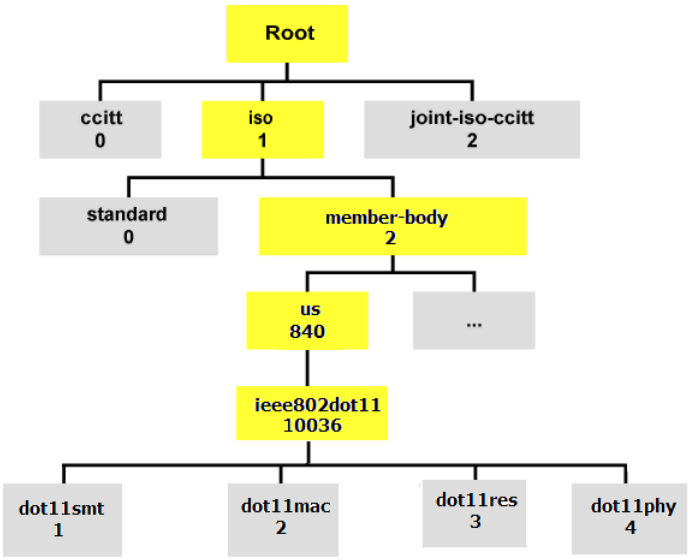

Figura 4.1: Situación en el árbol general de la MIB ieee802dot11

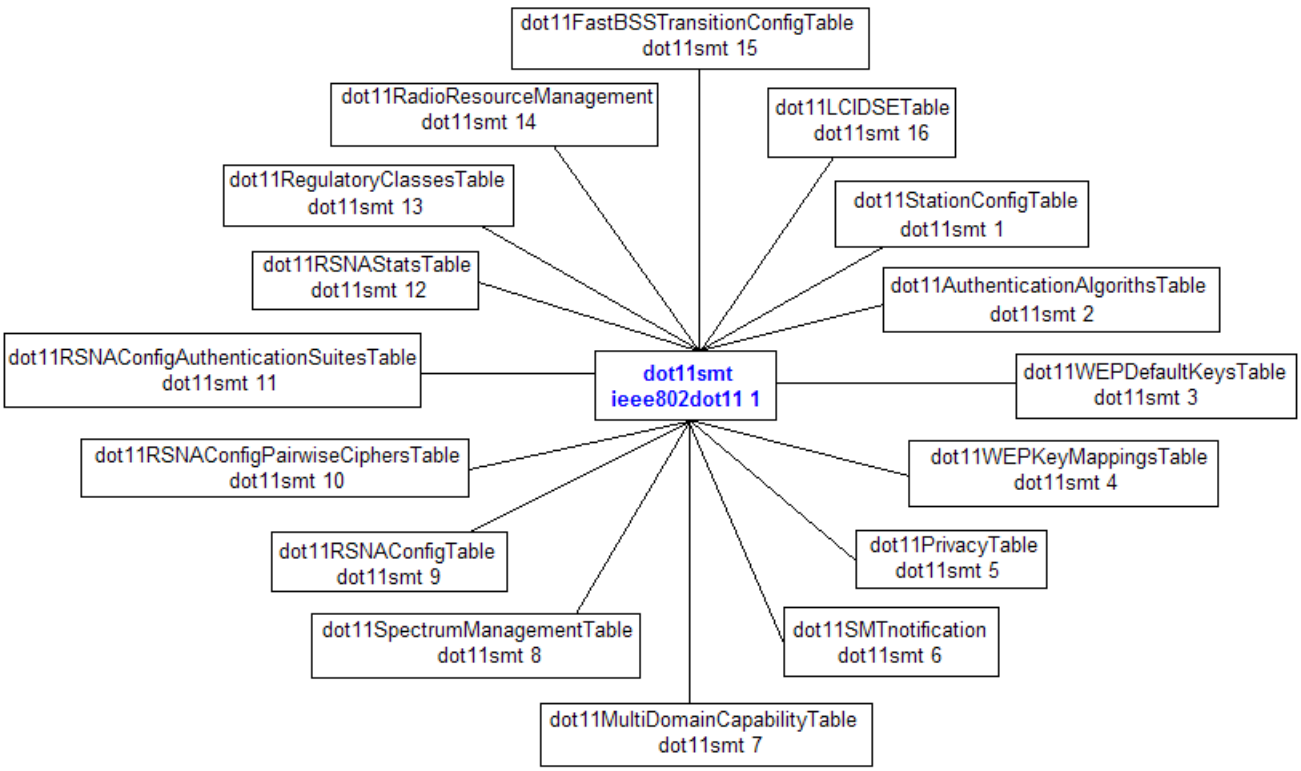

Figura 4.2: Subárbol a partir del nodo dot11smt

dot11mac Estos parámetros representan todos los factores que son necesarios en el control de acceso, generación y verificación de secuencias, así como en la validación de la entrega de los datos a capas superiores.

En la Figura 4.3 pueden verse los nodos que cuelgan de este nodo principal y que se corresponden con *tablas* en las que se incluyen los diferentes parámetros.

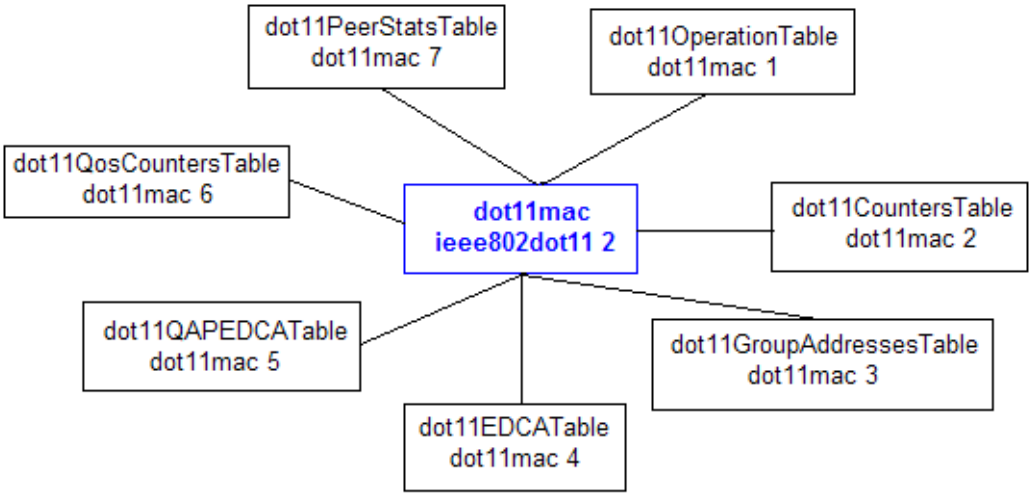

Figura 4.3: Subárbol a partir del nodo dot11mac

De todos estos parámetros los que van a interesar para el presente proyecto son básicamente *RTSThreshold*, *número de reenvíos antes del descarte del paquete*, *tamaño de fragmentación del paquete*, los cuales son hojas que cuelgan del nodo dot11mac 1, es decir, del dot11OperationTable, y *contadores que especifican el número de paquetes descartados*, los cuales son hojas que cuelgan del nodo dot11mac 2, es decir, dot11CountersTable.

dot11res Estos parámetros tienen que ver con el dispositivo en sí, es decir, nombre del producto, versión del mismo, etc. De este nodo el parámetro que me interesa viene especificado por defecto con el valor RTID (Resource Type ID managed object).

dot11phy Estos parámetros representan los factores que son necesarios para las operaciones físicas y que dependen del medio físico por lo que pueden variar entre estación y estación.

En la Figura 4.4, pueden verse los nodos que cuelgan de este nodo principal y que se corresponden con *tablas* en las que se incluyen los diferentes parámetros.

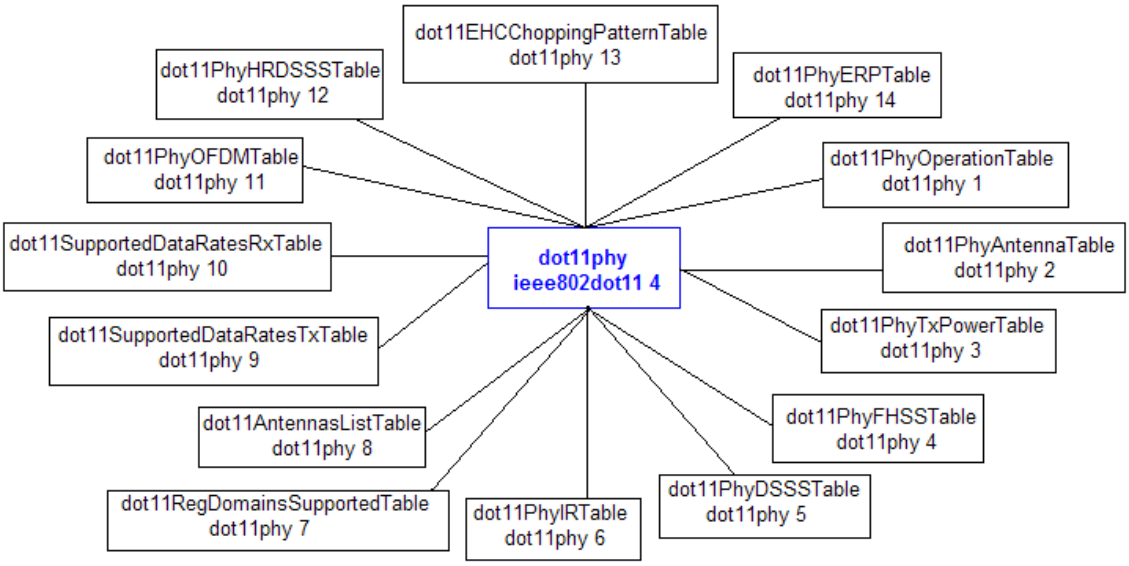

Figura 4.4: Subárbol a partir del nodo dot11phy

De todos estos parámetros los que van a interesar para el presente proyecto son básicamente la *frecuencia o canal*, la cual es una hoja que cuelga de dot11phy 5, es decir, del nodo dot11PhyDSSSTable, la *diversidad, rxantenna y txantenna*, los cuales son hojas que cuelgan de dot11phy 8, es decir, del nodo dot11AntennasListTable y la *velocidad de transmisión*, la cual es una hoja que cuelga de dot11phy 9, es decir, del nodo dot11SupportedDataRatesTxTable.

WEHAS-MIB

Esta MIB ha tenido que ser escrita durante el proyecto para complementar a la MIB del IEEE con los parámetros que faltaban y se necesitaban. Se ha situado en la siguiente posición dentro del árbol estándar:

iso(1).identified organization(3).dod(6).internet(1).experimental(3).wfehas(6789)

Como puede observarse en la Figura 4.5, está subdividida en dos campos principales que se explicarán a continuación.

IwConfig Los parámetros que cuelgan de este nodo van a ser todos los parámetros que se obtienen al ejecutar el comando *iwconfig* y que no tienen una definición en la MIB del ieee802dot11.

Por lo tanto de este nodo saldrán las hojas relativas al *estándar WiFi utilizado, el modo, la potencia de transmisión, la sensibilidad, la calidad del enlace, el nivel de señal, el nivel de ruido*, así como una *serie de contadores con el número de paquetes que se han descartado por diferentes razones*.

ProcSysDevWifi Los parámetros que cuelgan de este nodo van a ser todos los parámetros que tienen un valor en la ruta */proc/sys/dev/wifix* y que no tienen una definición en la MIB del ieee802dot11.

Por lo tanto de este nodo saldrán las hojas relativas al *ackrate, aktimeout, countrycode, ctstimeout, debug, fftxqmin, intmit, ledpin, regdomain, slottime, softled, TKIPMic, txinterperiod, XRPollCount y XRPollPeriod*.

Cabe destacar que, una vez escrita la MIB hay que comprobar que se ha hecho correcta-

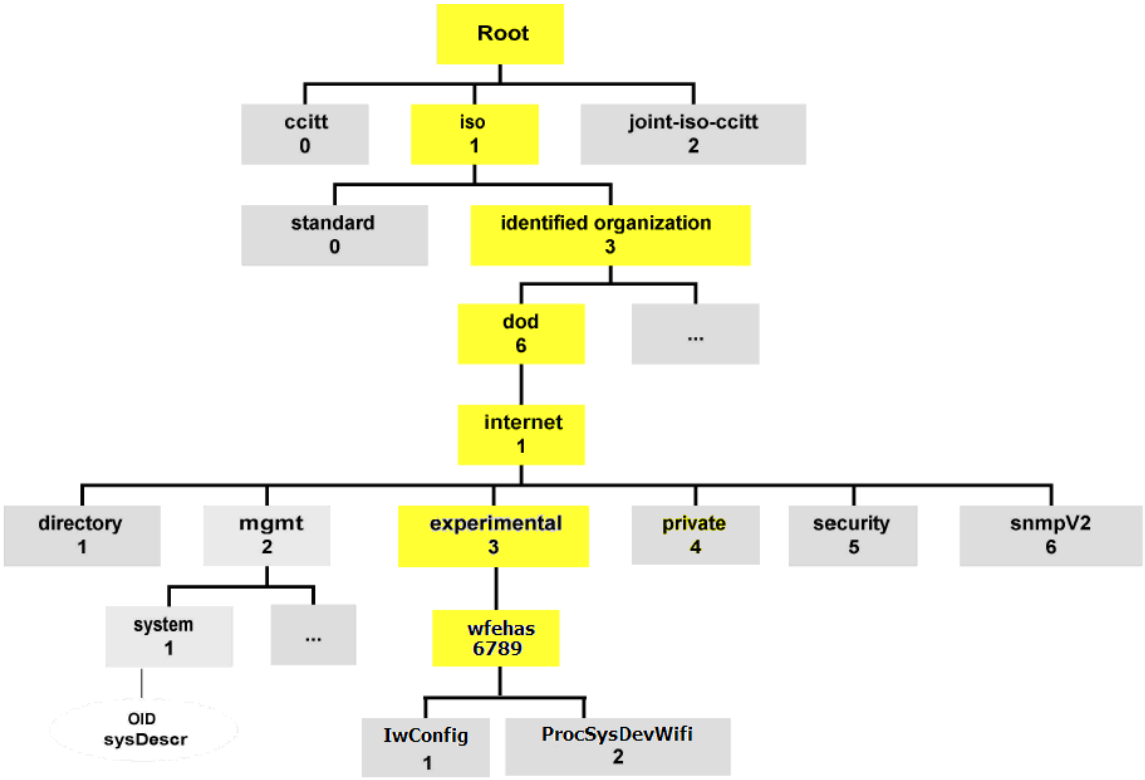

Figura 4.5: Situación en el árbol general de la MIB WEHAS

mente. Para realizar esta comprobación lo mejor es utilizar el programa *mib-browser* [19], el cual y, gracias a una interfaz gráfica, muestra el árbol con sus nodos y sus hojas si se ha descrito correctamente según se ve en la Figura 4.6, si no te muestra los errores correspondientes.

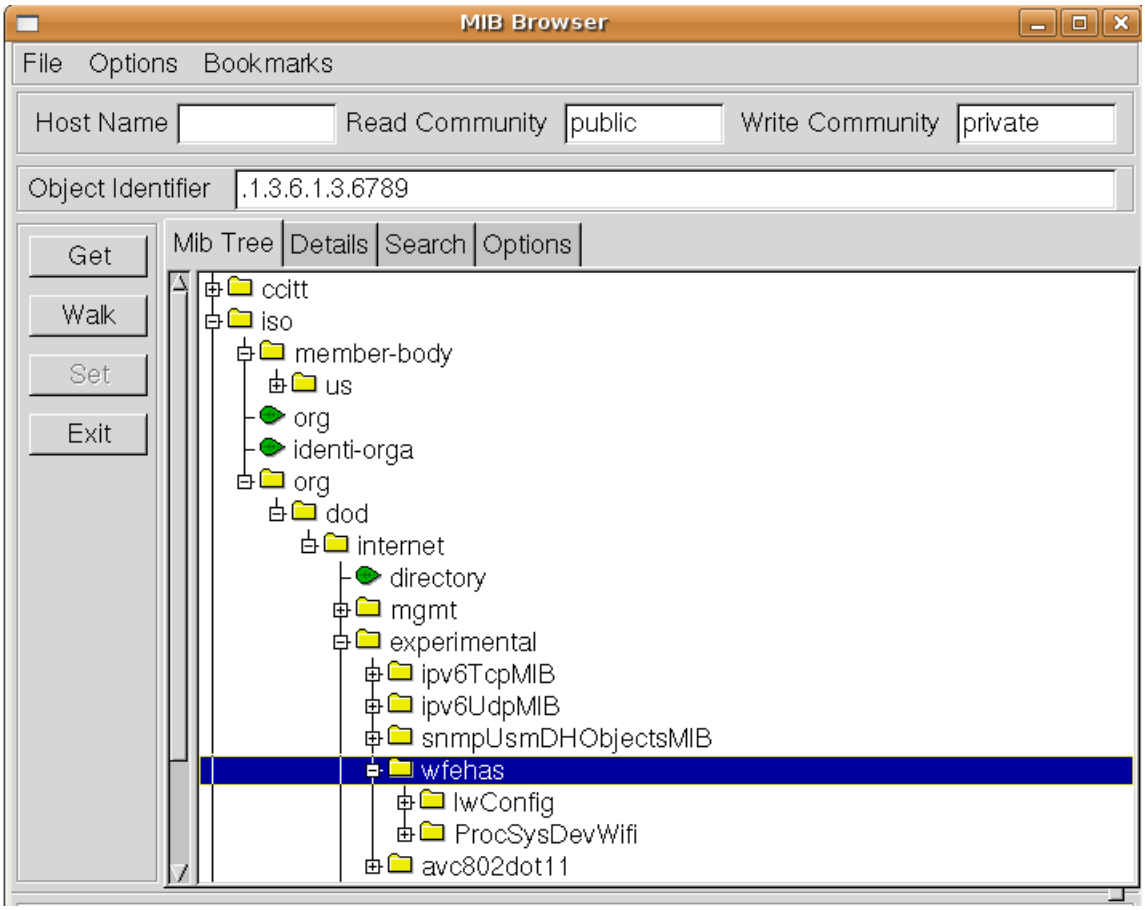

Figura 4.6: Imagen del mib-browser

No obstante, también puede cerciorarse la MIB utilizando el comando *snmptranslate* en una distribución Linux. Este comando devuelve la ruta, bien numérica, bien semántica, dependiendo de la opción utilizada, en la que está situado el objeto que se pida siempre y cuando está bien descrito. Si no también salen la serie de errores correspondientes.

```
snmptranslate -Of NOMBRE-MIB::nodo_a_preguntar
```

Este comando devuelve la ruta semántica como puede observarse en la Figura 4.7.

```
snmptranslate -On NOMBRE-MIB::nodo_a_preguntar
```

Este comando devuelve la ruta numérica como puede observarse en la Figura 4.8.

Además, para que el agente Net-SNMP pueda reconocer la MIB, hay que situarla en la carpeta que éste tenga configurada para guardarlas. En el caso de la distribución Linux hay que copiar la MIB que quiera utilizarse en la carpeta */usr/share/snmp/mibs*, según figura en el fichero de configuración */etc/default/snmpd*.

4.2.2 FICHERO EJECUTABLE

Una vez que se tiene el fichero de texto, MIB, con la definición y posición de los parámetros que se requieran, situado en la carpeta que corresponda para poder ser reconocido por el agente, hay que escribir la *programación* oportuna para obtener un fichero ejecutable dinámico que enlaza con el agente y lo extiende, devolviendo el valor de todos los parámetros especificados. En este caso se han tenido que programar dos ficheros ejecutables, uno para la MIB del IEEE y otro para la MIB adicional a la del IEEE desarrollada en el proyecto.

En cualquier caso, para obtener un fichero ejecutable dinámico, por un lado, se programan dos ficheros en lenguaje C. El fichero correspondiente a las cabeceras y estructuras utilizadas o fichero con extensión .h y el correspondiente a la implementación de las funciones necesarias para poder obtener el valor de los parámetros o fichero con extensión .c. Por otro lado, se programa un fichero Makefile, que será el encargado de crear a partir del fichero con extensión .c un fichero ejecutable dinámico o fichero con extensión .so, que es el que finalmente enlaza con el agente.

```
snmptranslate -Of WEHAS-MIB::wfehas
.iso.org.dod.internet.experimental.wfehas
```

Figura 4.7: Ruta semántica devuelta por el comando snmptranslate

```
snmptranslate -On WEHAS-MIB::wfehas
.1.3.6.1.3.6789
```

Figura 4.8: Ruta numérica devuelta por el comando snmptranslate

Ficheros en lenguaje de programación C

Como acaba de comentarse para cada fichero ejecutable se deben crear dos ficheros en lenguaje C: el fichero con extensión .h y el fichero con extensión .c. Estos ficheros, no obstante, tienen una estructura definida, ya que, al tener que enlazarse con el agente para ser una extensión del mismo, van a utilizar funciones propias de éste, así como una serie de estructuras determinadas. Esto implica que haya una herramienta que genera el esqueleto de estos ficheros, la llamada *mib2c*[20].

mib2c Esta herramienta está diseñada para coger una parte de árbol, a partir del nodo que se le especifique, y crear el esqueleto del código C con las funciones propias del agente para poder realizar la gestión de los parámetros de esa parte del árbol.

Hay que reseñar que lo que genera es sólo un esqueleto por lo que hay que complementar el código para que devuelva los valores que se quieran.

El esqueleto que genera, esto es, las funciones que serán utilizadas por el agente, no es único y puede ser configurable dependiendo del tipo de código y la utilidad que se le quiera dar al programa. De hecho, existen una serie de ficheros configurables por defecto de mib2c que pueden utilizarse e, incluso, pueden crearse ficheros configurables propios.

Los ficheros configurables que vienen por defecto son los siguientes:

- *mib2c.mfd.conf*: este fichero de configuración genera un esqueleto simple y fácil de utilizar. Suele usarse para principiantes. Pero el nodo origen a partir del que genera el código tiene que ser una tabla.
- *mib2c.scalar.conf* y *mib2c.int_watch.conf*: estos ficheros de configuración generan un esqueleto especialmente para el caso de querer valores escalares, es decir, valores que no formen parte de una tabla.
- *mib2c.iterate.conf*: este fichero de configuración genera un esqueleto especial para el caso en el que se quieran buscar un valor determinado dentro de una tabla.
- *mib2c.create-dataset.conf*: este fichero de configuración genera un esqueleto especial para el caso en el que no ocurra nada cuando se haga una función de set, es decir, una función de cambio de valor de algún parámetro.

- *mib2c.array-user.conf:* este fichero de configuración genera un esqueleto especial para el caso en el que se quiera centrar el programa en la función set, es decir, en el hecho de cambio de los parámetros.

- *mib2c.column_defines.conf* y *mib2c.column_enums.conf*: estos ficheros de configuración generan sólo las cabeceras para cada columna de las tablas o para los parámetros enumerados, respetivamente.

- *ucd-snmp style*: que es el elegido para llevar a cabo este proyecto, va a generar un esqueleto de carácter más general, independientemente del tipo de parámetros y válido tanto para la función set, es decir, cambio del valor de un parámetro, como para la función get, es decir, obtención del valor del parámetro.

Fichero con extensión .h Cuando mib2c genere este fichero, tan sólo va a contener las cabeceras de las funciones propias del agente, es decir, las funciones a las que llamará el agente para realizar las funciones que se le indiquen, bien sea *set*, cambio del valor, bien sea *get*, obtención del valor. Así como la función de inicialización de toda la parte del árbol que se le ha indicado.

Estas funciones son:

- `void init_nodo_principal(void)`

  Esta función inicializa toda la parte del árbol que se ha creado. Registrando los parámetros que forman parte de la MIB.

- `FindVarMethod var_tabla`

  Estas son las funciones a las que llama el agente cuando quiere obtener un valor determinado, es decir, *get*. En este proyecto los distintos parámetros se han distribuido en tablas, correspondientes a los nodos que cuelgan del nodo principal, de manera que habrá una función para cada tabla, la cuál estará compuesta por un número determinado de parámetros.

- `WriteMethod write_parámetro`

Estas son funciones a las que llama el agente cuando quiera cambiar el valor de un parámetro determinado, es decir, *set*. Va a haber una función para cada parámetro, independientemente de que el parámetro se encuentre en una tabla u otra.

Además de las cabeceras que se acaban de explicar, en este fichero y como parte complementaria al código creado por mib2c, se especifican las estructuras que serán utilizadas en el fichero con extensión .c y que se corresponden con los parámetros que forman la MIB. Como se ha expuesto anteriormente, de cada nodo principal cuelgan unas tablas correspondientes a los parámetros finales o nodos hojas. En este proyecto las estructuras creadas se corresponden con cada una de estas tablas o nodos padres a los nodos hojas. Por lo tanto, cada estructura está formada por los parámetros de una tabla. Así, al obtener cualquier valor, se rellena el elemento de la estructura correspondiente a ese parámetro. Además, en este fichero también se define cualquier constante global que haya que definir.

Fichero con extensión .c Cuando mib2c genera este fichero, por un lado, se va a crear el cuerpo de las funciones cuyas cabeceras se definieron en el fichero con extensión .h. Hay que tener en cuenta que este cuerpo está vacío y hay que rellenarlo con el código oportuno que calcule el valor de los parámetros que hay que devolver cuando estos se pidan.

Por otro lado, va a crear la estructura que se corresponde con la MIB a la que se hace alusión desde este código. Esta estructura especificará el nombre del parámetro, un valor, denominado *magic*, que representa al nombre del parámetro numéricamente y que hará más fácil la búsqueda del parámetro por parte del agente, así como el tipo de dato del parámetro, los permisos que tiene, es decir, si es de lectura, escritura, o ambos, su situación dentro del árbol, esto es especialmente importante para saber el OID que representa al parámetro y, la función a la que el agente debe llamar cuando quiera obtener su valor.

Por último, también se crea la función que se encarga de inicializar esta estructura en el agente para que tenga constancia de los parámetros de los que está compuesta la MIB así como de la función a la que tiene que llamar el agente para obtener el valor deseado.

Por lo tanto, el código que hay que completar, como se acaba de decir, va a ser básicamente la forma de obtener los valores de los distintos parámetros así como la estructura donde se guardan estos parámetros. No obstante, esto no será una labor tan fácil porque hay que tener presentes muchos aspectos.

4.2. IMPLEMENTACIÓN DE LOS PARÁMETROS

En primer lugar y como uno de los aspectos determinantes, hubo que tener en cuenta que puede haber más de una interfaz inalámbrica. A la hora de crear la estructura se considerá una buena opción crear una lista enlazada en la que cada nodo representase a una interfaz inalámbrica. De manera que, dependiendo del número de interfaces, esta lista se rellenase con más o menos nodos. El hecho de utilizar memoria dinámica es algo delicado y hay que tener cuidado tanto a la hora de pedir memoria como de liberarla. Así como a la hora del uso de los punteros. En la siguiente figura 4.9 puede verse el esquema general del programa cuando se hace la petición del valor de un parámetro.

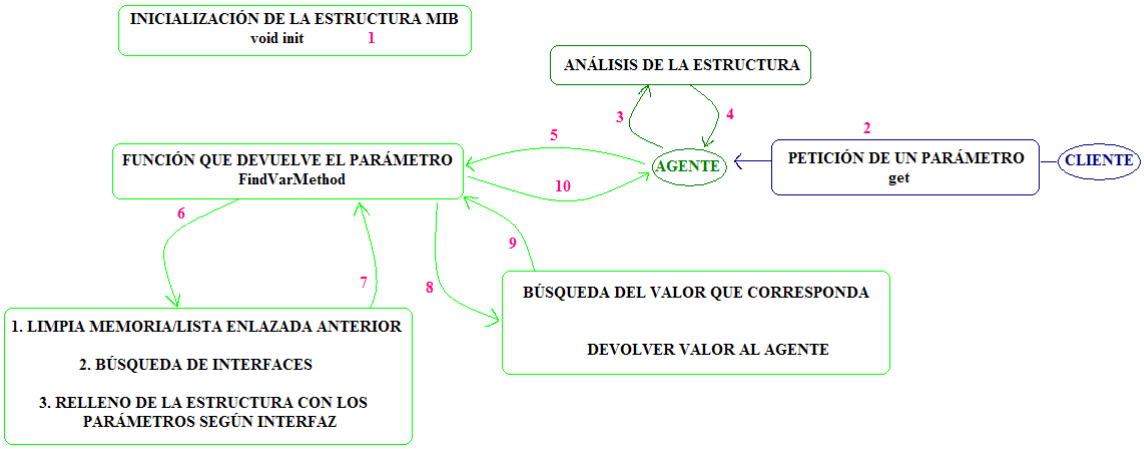

Figura 4.9: Esquema general del programa cuando le pide un valor el cliente al agente

Como puede observarse en primer lugar se inicializa la estructura para que el agente tenga el registro de los parámetros, su OID, su permiso, el tipo de datos y la función a la que debe llamar para obtener el valor de cada parámetro. Una vez que el cliente realiza una petición, el agente analiza la estructura y ve a qué función tiene que llamar. Esta función se ha programado para que limpie la memoria, es decir, la lista enlazada que hubiese creada, si es que estaba creada, busque todos los interfaces que haya y rellene la lista con los valores que le correspondan a cada interfaz, creando un nodo por interfaz. Cabe destacar que la lista enlazada en realidad no se rellena cada vez que se haga una petición. Ya que normalmente se pide más de un valor y para conseguir eficiencia, esta lista se rellena cada cierto tiempo. Por lo tanto, lo primero que hay es comprobar si ha transcurrido ese tiempo y si sí lo ha hecho se libera la memoria y se vuelve a crear la lista pero si no, se utiliza la lista que ya estaba creada. Cuando están todos los valores calculados, se busca el que se quiere devolver y se

devuelve al agente.

Cálculo de los valores Para realizar el cálculo de los valores lo primero que ha habido que tener en cuenta es cómo la computadora devolvía estos valores, ya que esa es la forma de calcularlos.

Para ver los pasos que seguía se ha utilizado el comando *strace*, el cual hace una traza de todas las llamadas que se realizan al sistema cuando ejecutamos un comando.

Tras el análisis de esta traza se puede concluir que la mayor parte de los valores se obtienen mediante *ioctl(input/output control)*. Las ioctl son unas llamadas al sistema en Unix [21] que permiten a una aplicación controlar o comunicarse con un driver de dispositivo. Una vez estudiadas y entendidas todas las ioctl que se necesitaban para el cálculo de los valores de los parámetros requeridos y, por intentar hacerlo lo más mantenible posible, se estudiaron las funciones que forman la librería *wireless-tools*[23], funciones que finalmente se han optado por utilizar.

Fichero ejecutable dinámico

Una vez completado todo el código necesario es imprescindible generar a partir de él y gracias a un Makefile, el fichero ejecutable dinámico con la extensin .so. Este fichero es el que realmente se cargar con el agente, por lo tanto, hay que especificarle al agente dónde debe encontrarlo. Para ello se añade en el fichero de configuración que se encuentra en */etc/snmp/snmpd.conf*, en distribuciones Linux, el camino donde estará mediante la siguiente línea: *dlmod nodo_principal camino_donde_encontrar_el_fichero_.so.*

No obstante, aunque se ha optado por esta forma existen tres maneras diferentes de integrar una MIB adicional en un agente[24].

1. El agente puede invocar a un script externo para que le devuelva la información necesaria.

2. El agente puede pasar la petición a otro agente o subagente que también implemente la MIB en cuestión.

3. Implementar el código necesario e incluirlo con el agente, que ha sido la opción elegida.

La tercera opción es la elegida porque se considera la más óptima ya que extiende al agente sin depender de otros factores externos.

4.2.3 FUNCIONES UTILIZADAS

Funciones desarrolladas

- `void beginning(void(*)(int, char *, struct wireless_info *));`

  Esta procedimiento, que tiene como argumento de entrada una función que va a corresponderse con un load_xxx(int, char *, struct wireless_info *), va a buscar todas las interfaces inalámbricas existentes y va a llamar a la función load_xxx correspondiente para que calcule los valores propios de cada interfaz y cree un nodo nuevo en una lista simplemente enlazada correspondiente al interfaz y sus valores.

- `void load_xxx(int, char *, struct wireless_info *);`

  Este procedimiento se encarga de calcular todos los valores correspondientes a su tabla y de crear la lista simplemente enlazada. Como argumentos de entrada tiene un entero, correspondiente al socket que se utilizará para calcular los valores, una cadena de caracteres que se corresponde con el nombre de la interfaz y la estructura wireless_info, la cual se rellenar con los parámetros que se vayan calculando.

- `void cleaning_xxx(struct xxx *);`

  Este procedimiento se encarga de limpiar memoria, deshaciendo la lista simplemente enlazada existente e inicializando los punteros correspondientes a NULL. Como argumento de entrada tiene el puntero al primer elemento de la estructura.

- `void clean();`

  Este procedimiento va a encargarse de inicializar todos los punteros que son globales y que van a utilizarse en el programa para la creación de la lista simplemente enlazada y todas las funciones que tengan que ver con rellenarla, recorrerla y limpiarla.

- `int freq2channel(double);`

  Esta función va a encargarse de calcular el canal que le corresponde a una frecuencia determinada según la Tabla 4.2 y la Tabla 4.3. Como argumento de entrada se encuen-

tra el valor de la frecuencia y como argumento de salida se obtiene un entero que se corresponde con el canal.

- `int get_value();`

  Esta función va a devolver el valor entero xxx que leer en el fichero /proc/sys/dev/wifiX/xxx.

- `void set_value();`

  Este procedimiento va a escribir el valor entero xxx en el fichero /proc/sys/dev/wifiX/xxx

- `int time_pass();`

  Esta función va a calcular el tiempo transcurrido entre la petición de un parámetro y otro. Si el valor que devuelve es 1 quiere decir que ha pasado el tiempo suficiente (estimado en 5 segundos) como para volver a calcular la lista simplemente enlazada. Mientras que 0 sigue devolviendo los valores que se calcularon en la lista enlazada existente.

Funciones propias del agente

- `void init_xxx(void);`

  Este procedimiento se encarga de inicializar todos los parámetros propios de la MIB, de manera que queden registrados en el agente.

- ```
  void register_mib(xxx, xxx_variables, tipo_estructura,
                    xxx_variables_oid);
  ```

  Este procedimiento, el cual forma parte de la función anterior *init_xxx*, es el que registra la MIB especificando su posición dentro del árbol (oid) y los parámetros que la forman incluyendo el tipo de parámetro, el tipo de acceso, esto es, si es lectura o escritura o ambos, la función a la que hay que llamar para su cálculo y un valor representativo. Como argumentos de entrada tenemos el oid y la estructura, además del nombre del primer nodo.

- ```
  unsigned char * var_xxx(struct variable *vp, oid *name,
                          size_t *lenght, int exact, size_t *var_len,
                          WriteMethod **write_method);
  ```

Esta es la función a la que llama el agente cuando quiere hacer una petición de algún valor específico. Para ello le pasa una *estructura vp*, la cual incluye parámetros determinantes como el *oid* del valor que quiere obtener, así como el número correspondiente al parámetro concreto, es decir, *magic*, el parámetro *exact*, el cuál especifica si el oid es específico o bien se refiere a la devolución de más de un parámetro y *WriteMethod*, que es la función que, entre otras cosas, especifica qué hacer en caso de fallos. Se devuelve un puntero a un unsigned char, es decir, al primer byte donde se encuentra el valor a devolver.

- `int snmp_oid_compare(const oid *, size_t, const oid *, size_t);`

  Esta función devuelve un valor positivo si el primer oid es mayor que el segundo, un valor nulo si son iguales y un valor negativo en el caso en el que el primer oid sea menor que el segundo. Como argumentos de entrada se encuentran los oid correspondientes que se van a comparar así como la longitud de los mismos.

- ```
  int write_xxx(int action, u_char *var_val, u_char var_val_type,
                size_tvar_val_len, u_char *statP, oid *name,
                size_t name_len);
  ```

  Esta función tiene varios cometidos. Aunque principalmente se utilizará para realizar la petición de set, es decir, la modificación de algún parámetro.

- `nametoindex(char * ifName);`

  Esta función convierte la cadena del nombre de la interfaz en un número natural

Funciones del sistema

- `int ioctl(sock, SIOCGIWNAME, &wrq);`

  Esta función se encarga de ponerle nombre a la estructura wrq que es del tipo iwreq para posteriormente poder ir rellenándola segn las consiguientes ioctl. Devuelve un valor positivo o nulo si se rellena el nombre de manera correcta mientras que devuelve un valor negativo en caso de no rellenarse el nombre correctamente. Como argumentos de entrada tiene el entero correspondiente al socket, que es imprescindible para poder usar las diferentes ioctl, la estructura que se acaba de explicar y el nombre específico de la ioctl a la que se llama.
  ```

Funciones de la librería wireless tools

- `int iw_sockets_open(void);}`

  Esta función calcula un entero que se corresponde con el socket que va a utilizarse.

- `int iw_get_ext(sock, ifName, IOCTL,&wrq);`

  Esta función viene a ser una ioctl pero donde, además de pasarle el socket, la estructura de tipo iwreq y la función ioctl especfica, se la pasa el nombre del interfaz. Como todas las funciones ioctl, devuelve un valor positivo o nulo en caso de completar la ioctl con éxito, mientras que devuelve un valor negativo en caso de que no se pueda realizar la obtención del parámetro. En este caso se utilizarn las siguientes IOCTL:

  - SIOCGIWAP obtiene la dirección MAC del AP.
  - SIOCGIWRTS obtiene el tamaño en bytes del paquete que usa el protocolo RTS/CTS.
  - SIOCGIWRETRY obtiene el tiempo en el que se intenta reenviar el paquete hasta que se descarta.
  - SIOCGIWFRAG obtiene el tamaño del paquete que se fragmenta.
  - SIOCGIWRATE obtiene la velocidad binaria (bps) por defecto.
  - SIOCGIWTXPOW obtiene la potencia transmitida en dBm.
  - SIOCGIWSENS obtiene la sensibilidad en dBm.

- `int iw_get_basic_config(sock,ifName, &(wi->b));`

  Esta función obtiene toda la configuración básica wireless del driver completando la estructura wireless_config, devolviendo un número positivo o nulo si se obtiene sin problemas y devolviendo un número negativo en caso de algún problema. Como argumentos de entrada se encuentran el socket, el nombre del interfaz y la estructura wireless_config, que viene a ser una subestructura de la estructura wireless_info.

- `int iw_get_stats(sock, ifName,(wi->stats), &(wi->range),(wi->has\_range));`

  Esta función rellena el campo stats de la estructura wireless_config con una serie de estadísticas wireless del driver. Devuelve un número positivo o nulo en caso de que se

pueda rellenar la estructura con éxito, mientras que devuelve un valor negativo en caso en el que la estructura no pueda rellenarse con éxito.

Capítulo 5

Pruebas

En este capítulo va a hacerse una descripción de las pruebas realizadas así como del escenario utilizado para las mismas.

Aunque, en realidad, para las pruebas sobre el correcto funcionamiento del programa implementado tan solo se hubiesen necesitado unos routers, se ha optado por implementar todo el SGR incluyendo también la herramienta de gestión, para comprobar cómo se integra el programa implementado con la misma.

5.1 ESCENARIO UTILIZADO

El esquema del escenario puede verse en la Figura 5.1 y representa la realidad de la red en la que va a ser utilizado, es decir, la red del río Napo B. En concreto se representa la conexión entre tres nodos de la red real que se sitúan en 10.0.0.0/8.

Por una lado tenemos el *receptor*, que se corresponde con el primer nodo de la red real, luego el *repetidor1* y *repetidor2*, que se conectan mediante cable y no de forma inalámbrica, que representan el segundo nodo de la red real y, el *emisor*, que representa el tercer nodo de la red real y que se conecta de forma inalámbrica con otro router el cuál está conectado a dos PC simulando lo que sería la casita azul.

Lo routers principales son *WRAP* [26], con tarjetas *Atheros* y compact flash donde se ha cargado el SO *Voyage versión 0.5* [1]. Mientras que el router que se corresponde con casita azul en un *Linksys* [27].

[1] Voyage es una SO basado en Linux pero muy reducido con el fin de ocupar poco espacio

En el esquema puede verse como el receptor se conecta mediante switch a dos servidores en la subred 192.168.120.0/24. Por un lado, al servidor *Asterisk*, el cual sirve para llamadas telefónicas y, por otro lado, al servidor con la dirección 192.168.120.135, el cuál proporcionará la salida a Internet de la red y hace la función de *herramienta de monitorización*.

Cabe señalar que el encaminamiento se ha hecho de forma dinámica mediante OSPF *(Open Shortest Path First)*.

5.2 SOTFWARE NECESARIO

En el servidor central la herramienta de monitorización que se ha instalado ha sido *Centreon* [28], que es un configurador web que opera sobre *Nagios* [29].

La forma en la que se ha configurado la misma puede encontrarse en [30].

Además de la herramienta de monitorización, dentro del servidor central hay que instalar el *agente snmp*, que se corresponde con el *paquete snmpd*, el cuál se encuentra en los repositorios de Linux. Es importante instalar el agente para poder monitorizar también el servidor.

Por otro lado también hay que instalar el *cliente snmp*, que se corresponde con el *paquete snmp*, el cuál también se encuentra en los repositorios de Linux. Este paquete es especialmente importante porque el cliente es el que se encarga de pedir la información al resto de los agentes instalados en los demás nodos.

Y también hay que incluir todas las *MIB* que vayan a ser utilizadas para que se puedan reconocer todos los parámetros que se pidan. En este caso se han incluido tanto la MIB del IEEE como la MIB adicional y complementaria a la del IEEE creada.

Por otro lado, en los sistemas gestionados, en este caso el resto de los routers que forman la red, hay que instalar el *agente snmp*, que como se ha dicho se corresponde con el *paquete snmpd* el cuál puede bajarse de los repositorios Linux. Además de incluir tanto la *MIB* como el *ejecutable dinámico* implementado que se enlaza con el agente para que pueda obtener el valor de los parámetros que se le pidan. En este caso se han incluido las dos MIB, tanto la del IEEE como la creada adicional a ésta, además de los ficheros ejecutables dinámicos programados y correspondientes a la MIB del IEEE y la MIB complementaria, respectivamente.

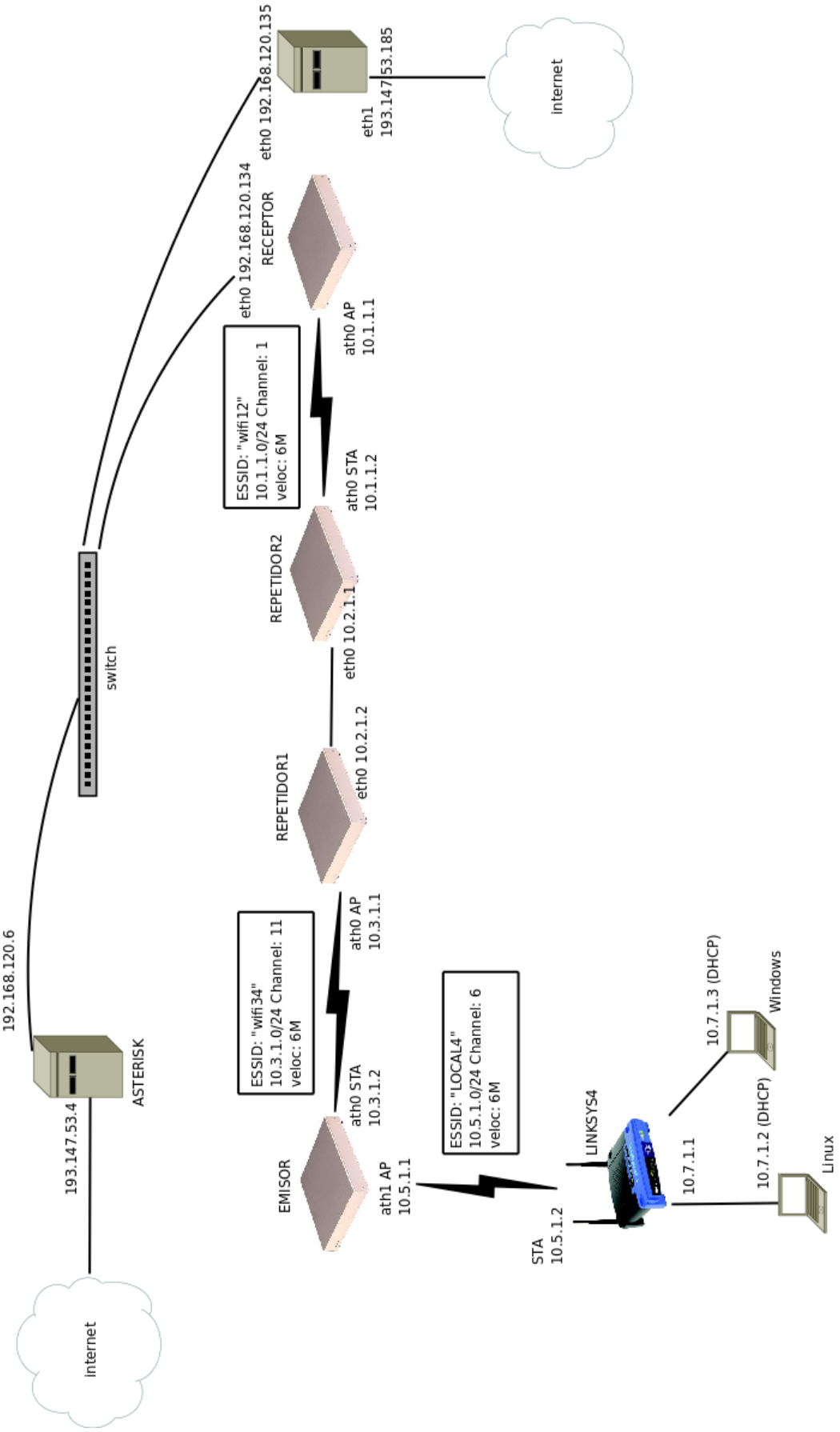

Figura 5.1: Esquema general del escenario de prueba

Sobre esto cabe decir que, como ya se ha comentado en alguna ocasión, las MIB van a guardarse en la carpeta que incluye todas las mibs, es decir, en */usr/share/snmp/mibs*.

Por otro lado, los ejecutables dinámicos vamos a ponerlos en */usr/local/lib*.

Cabe destacar que para poder realizar la gestión hay un par de puntos importantes que hay que configurar. Por un lado, hay que especificar la comunidad y el tipo de permisos que esta tiene. La comunidad viene a ser el dato que especifica el nombre de la entidad que tiene acceso para poder pedir o modificar parámetros. Estos permisos también hay que especificarlos porque puede tener acceso de sólo lectura, de sólo escritura o de lectura y escritura.

Todo esto puede configurarse en el fichero *snmpd.conf*, el cual se encuentra en */etc/snmp/snmpd.conf*.

En este caso se ha creado la comunidad *Ehas* la cual tiene permisos tanto de lectura como de escritura.

Por otro lado, hay que destacar que hay que darle permiso al agente para pueda escuchar peticiones que le vengan desde otro dispositivo que no sea el host local. Esto se configura en el fichero */etc/default/snmpd*. Además, en este mismo fichero, hay que darle permiso de sudo al agente para que pueda acceder a ciertos ficheros necesarios para el cálculo de los valores.

Por último recordar que el SO del servidor central es un *Ubuntu versión 9.04* mientras que en las compact flash de los routers, como ya se ha comentado, se ha instalado una *Voyage versión 0.5*.

5.3 CONFIGURACIÓN EN CENTREON

Una vez instalados todos los componentes necesarios descritos en el apartado anterior, y teniendo en cuenta que básicamente la herramienta de configuración le pide al cliente que éste le pida al agente un parámetro concreto, lo que hay que configurar en el Centreon-Nagios es el parámetro snmp que se quiera monitorizar.

Para poder hacer esto se utiliza un plugin que se llama *check_snmp*.

Entre los parámetros que hay que definir se encuentran el OID, la comunidad, la versión.

Hay que reseñar que la compact flash, en la que se ha instalado el SO Voyage versión 0.5, tiene un número limitado de lecturas y escrituras. Por lo tanto, normalmente, para evitar que se escriba de manera incontrolada, que suele ser lo más delicado, se encuentra por defecto en

estado *remountro*, es decir, en estado de sólo lectura. En el caso en el que se quiera escribir, modificar valores, hay que ponerla en estado *remountrw*, es decir, en estado tanto lectura como escritura. De no estar en este estado, el agente snmp se parará y dejará de funcionar, por lo que habrá que volver a relanzarlo.

Como último destacar que la prueba principal para saber si el agente es capaz de devolver los valores pedidos se hace con los comandos *snmpget, snmpset o snmpwalk* [2]dependiendo de si se quiere obtener un sólo valor, modificar un valor o hacer un recorrido desde un nodo especificado. En estos comandos hay que especificar tanto la comunidad como la versión, así como el nodo al que se le quiere hacer la petición. En las Figura 5.2, Figura 5.3, Figura 5.4, Figura 5.5 y Figura 5.6 pueden verse unos ejemplos de los valores devueltos al ejecutar estos comandos.

```
snmpwalk -c Ehas -v 2c localhost IEEE802dot11-MIB::ieee802dot11
IEEE802dot11-MIB::dot11StationID.3 = STRING: 0:7:40:b1:79:21
IEEE802dot11-MIB::dot11DesiredSSID.3 = STRING: "EHAS"
IEEE802dot11-MIB::dot11RTSThreshold.3 = INTEGER: 2304
IEEE802dot11-MIB::dot11ShortRetryLimit.3 = INTEGER: 7
IEEE802dot11-MIB::dot11FragmentationThreshold.3 = INTEGER: 2346
IEEE802dot11-MIB::dot11FailedCount.3 = Counter32: 0
IEEE802dot11-MIB::dot11WEPUndecryptableCount.3 = Counter32: 0
IEEE802dot11-MIB::dot11ResourceTypeIDName.0 = STRING: RTID
IEEE802dot11-MIB::dot11CurrentChannel.3 = INTEGER: 11
IEEE802dot11-MIB::dot11SupportedTxAntenna.3 = INTEGER: false(2)
IEEE802dot11-MIB::dot11SupportedRxAntenna.3 = INTEGER: false(2)
IEEE802dot11-MIB::dot11DiversitySelectionRx.3 = INTEGER: 0
IEEE802dot11-MIB::dot11SupportedDataRatesTxValue.3 = INTEGER: 108000
```

Figura 5.2: Resultado del comando snmpwalk para la MIB IEEE802dot11

```
snmpget -c Ehas -v 2c localhost IEEE802dot11-MIB::dot11StationID.3
IEEE802dot11-MIB::dot11StationID.3 = STRING: 0:1a:2b:6:1d:1f
```

Figura 5.3: Resultado del comando snmpget para la MIB IEEE802dot11

[2]Estos comandos son propios del cliente. Para poder utilizarlos tiene que estar instalado el paquete del cliente snmp

```
snmpwalk -c Ehas -v 2c localhost WEHAS-MIB::wfehas
WEHAS-MIB::Name.3 = STRING: "IEEE 802.11g"
WEHAS-MIB::Mode.3 = INTEGER: managed(2)
WEHAS-MIB::TxPower.3 = INTEGER: 20
WEHAS-MIB::Sensitivity.3 = INTEGER: 8
WEHAS-MIB::LinkQuality.3 = INTEGER: 91
WEHAS-MIB::SignalLevel.3 = INTEGER: -38
WEHAS-MIB::NoiseLevel.3 = INTEGER: -86
WEHAS-MIB::InvalidNWID.3 = INTEGER: 0
WEHAS-MIB::InvalidFRAG.3 = INTEGER: 0
WEHAS-MIB::InvalidMISC.3 = INTEGER: 0
WEHAS-MIB::MissedBeacon.3 = INTEGER: 0
WEHAS-MIB::ACKRate.3 = INTEGER: normal(0)
WEHAS-MIB::ACKTimeout.3 = INTEGER: 0
WEHAS-MIB::CountryCode.3 = INTEGER: 0
WEHAS-MIB::CTSTimeout.3 = INTEGER: 0
WEHAS-MIB::Debug.3 = INTEGER: 0
WEHAS-MIB::FFTxQmin.3 = INTEGER: 0
WEHAS-MIB::IntMit.3 = INTEGER: 0
WEHAS-MIB::LedPin.3 = INTEGER: 0
WEHAS-MIB::RegDomain.3 = INTEGER: 0
WEHAS-MIB::SlotTime.3 = INTEGER: 0
WEHAS-MIB::SoftLed.3 = INTEGER: 0
WEHAS-MIB::TKIPMic.3 = INTEGER: 0
WEHAS-MIB::TxInterPeriod.3 = INTEGER: 0
WEHAS-MIB::XRPollCount.3 = INTEGER: 0
WEHAS-MIB::XRPollPeriod.3 = INTEGER: 0
```

Figura 5.4: Resultado del comando snmpwalk para la MIB WEHAS

```
snmpget -c Ehas -v 2c localhost WEHAS-MIB::Name.3
WEHAS-MIB::Name.3 = STRING: "IEEE 802.11g"
```

Figura 5.5: Resultado del comando snmpget para la MIB WEHAS

```
snmpget -c Ehas -v 2c localhost WEHAS-MIB::Mode.3
WEHAS-MIB::Mode.3 = INTEGER: managed(2)
```

Figura 5.6: Resultado del comando snmpget para la MIB WEHAS

Capítulo 6

Redes virtuales para gestión

La idea básica que se busca al crear una VPN *(Virtual Private Network, Red Privada Virtual)* es poder acceder a los dispositivos de una red **A** desde otra red **B** a través de Internet, según el esquema de la Figura 6.1, como si las dos redes formasen una única red privada.

Figura 6.1: Situación inicial

Esto es especialmente importante cuando dispositivos de redes que no se encuentran físicamente unidas tienen que trabajar de forma conjunta. En el caso en cuestión de la red del Napo y, en concreto para el SGR, el problema planteado viene cuando quiere ponerse el nodo gestor en un dispositivo exterior a la red. En este momento los dispositivos de la red no podrían ser direccionados por el nodo gestor, ya que estarían en una red privada e innaccesible y, quedarían sin gestionar. De ahí la importancia del estudio de la VPN.

Existen diversas maneras de conectar dos redes separadas físicamente:

1. Líneas dedicadas. Esta opción es cara y no es siempre viable. Consiste en mantener unas líneas de forma permanente entre los puntos que quieran.

2. VPN. Cuando se conectan dos redes remotas por medio de Internet, el principal proble-

ma viene derivado de la seguridad. Por lo tanto, la idea, es proporcionar a los paquetes enviados esta seguridad que falta de manera que sólo los dos extremos sean capaces de acceder a esos datos creando un *túnel*. Para ello es necesario garantizar:

- Autenticación y control de acceso. Es decir, asegurar que la persona que accede a la red es quién dice ser, así como los poderes que le corresponden.
- Integridad. Este factor se refiere a la información, es decir, garantizar que los datos que se envían no hayan sido alterados.
- Confidencialidad. Es decir, garantizar que nadie intermedio pueda acceder a ellos.

Cuando se proporcionan estos factores a los paquetes enviados entre un extremo y otro se dice que se ha creado un túnel entre ambos.

El estudio de este capítulo va a centrarse en los túneles. En concreto, entre las diversas herramientas que existen para crear túneles se han seleccionado: pptp(*Point-to-Point Tunneling Protocol*) [31], IPsec [32] y OpenVPN [33].

6.1 PPTP

Es un protocolo creado por Microsoft que permite el envío seguro de datos a través de Internet.

Es una extensión del protocolo ppp (*Point-to-Point*) [34]. Y consiste en encapsular el paquete ppp sobre datagramas IP (*Internet Protocol*), gracias al protocolo GRE *(General Routing Encapsulation)* Las direcciones IP del cliente y servidor pptp son dinámicas y se asignan mediante NCP *(Network Control Protocol)*, durando la sesión.

Los paquetes enviados se cifran y autentican basándose en los mismos mecanismos que ppp. Respecto al cifrado, se utiliza un *algoritmo hash*, función cuya entrada es un gran conjunto de información y cuya salida es un conjunto finito menor al de la entrada llamado huella dactilar. Es un método unívoco y robusto, que se valdrá de la contraseña del usuario para crear la huella dactilar, mientras que para autenticar utilizará *pap*, método de autenticación por usuario y contraseña, y *chap*, método más complejo pero más seguro. El que quiere autentificar envía una cadena aleatoria al que se quiere autentificar, junto a su nombre de ordenador. El que quiere autentificarse utiliza el nombre del ordenador para buscar

la clave apropiada, la combina con la cadena aleatoria y utiliza una función de codificación para enviar la información al destino. El destino, que quiere autentificar, recibe esta codificación junto con el nombre del ordenador del que le envía la información. El destino realiza la misma computación y quedará autentificado si obtiene el mismo resultado.

Destacar que los puntos positivos de este protocolo son:

1. Fácil de utilizar.
2. Disponible en muchos routers.
3. Encapsular a nivel 2, por lo tanto, es capaz de soportar protocolos no IP.

Siendo el principal inconveniente el siguiente:

- Tiene problemas relativos a la seguridad. De hecho, ha sido rota.

6.2 OpenVPN

La idea de OpenVPN es la creación de una interfaz *tun* dentro del dispositivo origen y destino. Esto confiere una gran libertad y variedad para las configuraciones.

Puede implementar conexiones de capa 2 y capa 3, así como utilizar estándares de la capa de aplicación.

El kernel recibe las tramas que las aplicaciones de usuario le envían a través de la interfaz *tunX*, de igual modo que los programas de espacio de usuario reciben las tramas que el kernel les envía a través de este mismo interfaz. De esta manera no hay que tocar el kernel.

Para conferirle la seguridad a los paquetes, OpenVPN utiliza unas herramientas muy robustas tanto en cifrado como en autenticación.

Los puntos positivos de este protocolo son:

- Es muy seguro. Además las tecnologías de cifrado utilizadas son estandarizadas.
- Es sencillo de usar y de configurar.
- Está bien estructurado. Se basa en tecnología modular. No modifica el kernel.
- Compatible con firewall y proxies.

- No presenta problemas con NAT.
- Es SW libre.

No obstante su principal inconveniente es:

- No es estándar por lo que no está implementado en muchos routers.

6.3 IPsec

Lo primero que hay que destacar es que IPsec no es una herramienta exclusiva para crear túneles. De hecho, IPsec surgió para paliar las deficiencias que tenía IP en cuanto a la seguridad de los datos. No obstante, si se configura en el modo correcto puede utilizarse para crear túneles. IPsec actúa a *nivel 3*.

Antes de centrarnos en la tunelización por IPsec vamos a describirlo de forma general. Como comentamos al principio, IPsec va a tener diferentes modos de funcionamiento, en concreto dos: modo transporte y modo túnel. En el modo transporte sólo se cifran los datos. De esta manera, se garantiza una conexión segura entre dos host, pero sin crear un túnel entre ellos, sólo asegura una transferencia de información segura entre ambos. Sin embargo, el modo túnel cifra el paquete en su totalidad, convirtiendo el resultado en un nuevo paquete IP, con un origen y un destino (el cliente y el gateway), pudiendo crear los túneles, según se ve en la Figura 6.2. Esto tiene un problema y es que no es compatible con los NAT*(Network Address Translation)*. No obstante hay una solución a este problema y son las NAT-T (Traversal). Aunque no se ha estudiado ese problema en profundidad.

A partir de aquí supondremos que estamos en *modo tunel*.

Para garantizar la seguridad, IPsec utiliza AH *(Authentication Header)* o ESP *(Encapsulated Security Payload)*. Mientras que AH sirve sólo para autenticar, ESP además, cifrará, resolviendo además los problemas de confidencialidad.

El funcionamiento general sería el siguiente: en el extremo transmisor se va generando tráfico y se mira si ese tráfico cumple una serie de reglas o no. Si sí las cumple se cifra. Entonces se ponen de acuerdo los dos extremos sobre el algoritmo que van a utilizar y se crean las claves. Para hacer el intercambio de claves de utiliza el protocolo IKE *(Internet Key Exchange)* o ISAKMP*(Internet Security Association and Key Management Protocol)*. Otro

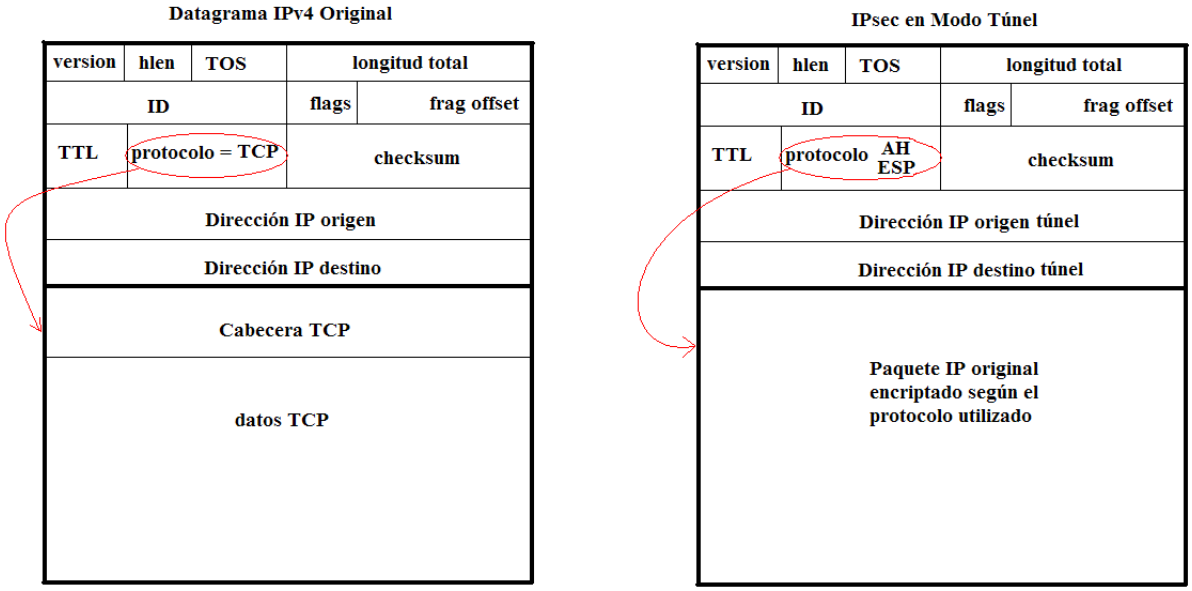

Figura 6.2: Esquema OpenVPN

protocolo de intercambio seguro es Diffie-Hellman, pero este se usará más en OpenVPN. Una vez hecho esto se le asocia un SA (Security Association). Este SA tiene tres parámetros principales: dirección destino, el índice del parámetro de seguridad SPI *(Security Parameters Index)*, que me apuntará a la clave de sesión y al tipo de algoritmo utilizado y el protocolo de seguridad (ESP o AH).

En el extremo receptor, cuando llega el paquete, lo primero que se mira es si tiene asociada una SA. Si no la tiene se descarta. Si sí que la tiene se busca en la base de datos mediante el SPI, que es el puntero que me especifica la clave de sesión y al algoritmo utilizado para poder descifrar. Una vez descifrado se compara con las reglas que deben cumplir el tráfico. Si no las cumple, el paquete se descarta y sólo llega a su destino en caso de que lo cumpla.

Las ventajas que presentan son:

- Es un estándar que se encuentra en muchos dispositivos.
- Es una tecnología probada y conocida.

No obstante, presenta una serie de inconvenientes:

- Necesita hacer modificaciones críticas del kernel.
- Es un protocolo complejo, tanto a la hora de configurar como la tecnología en sí.

- Tiene problemas con firewall y NAT.

6.4 RESULTADOS

En todos los casos el esquema de prueba utilizado es el representado en la Figura 6.3 y consta de un dispositivo origen conectado a Internet y que se corresponde con el *cliente*, un router Mikrotik [35], también conectado a Internet, el cual hace de *servidor*, tras el cuál se encontrará una red privada formada por otros dispositivos los cuales se van a gestionar.

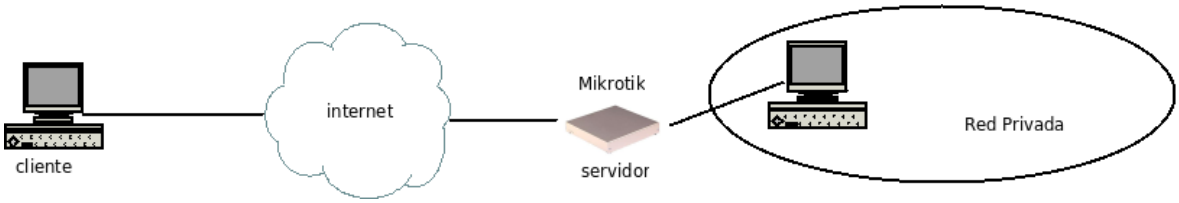

Figura 6.3: Esquema del escenario de prueba para VPN

Se han configurado los elementos del mismo según se explica en el Anexo C, para probar el funcionamiento de los túneles.

El resultado ha sido que, tanto para pptp como para OpenVPN se ha comprobado cómo el túnel se creaba, para lo cual se ha hecho un *ping* desde el cliente hasta el dispositivo de la red privada, accediendo al mismo sin problema. IPsec ha resultado demasiado complejo y no se ha conseguido crear el túnel.

Por otro lado y, sobre el mismo escenario de prueba, se ha querido probar cómo una herramienta de gestión instalada en el cliente era capaz de autodescubrir un dispositivo que se encontrase dentro de la red privada. Ya que, si la herramienta es capaz de descubrir el dispositivo, puede comunicarse con el mismo y, por lo tanto, gestionarlo, que es lo que buscamos.

Para esta prueba, la herramienta utilizada ha sido Zabbix [36].

Zabbix puede ser configurado para autodescubrir dispositivos según una serie de especificaciones. Lo que se ha hecho es instalar Zabbix en el cliente y configurarla para que autodescubriese cualquier dispositivo al que pudiese hacerle un *ping*. Una vez hecho esto se ha iniciado un túnel pptp entre el cliente y el servidor Mikrotik.

El resultado ha sido el esperado, ya que Zabbix ha sido capaz de autodescubrir el dispositivo que se encontraba en la red privada.

6.5 CONCLUSIONES

Como primera conclusión importante, se ha comprobado cómo es posible gestionar un dispositivo de una red privada mediante otro dispositivo situado en una red exterior y que accede a la misma por Internet mediante la creación de túneles.

Por otro lado, tras evaluar las herramientas pptp, IPsec y OpenVPN, se ha concluido que la mejor opción es usar OpenVPN, ya que:

- Aunque pptp presenta buenas características, su seguridad ha sido rota.
- IPsec se presenta demasiado complejo y no se ha conseguido terminar de configurar.
- OpenVPN, además de las ventajas expuestas salva el inconveniente de no estar implementado en muchos routers, ya que, los routers con los que se está empezando a trabajar en EHAS, de la marca Meshlium [37], sí que incluyen este protocolo.

Como último punto destacar que, un problema planteado respecto a este tema es el que surge en el caso en el que se quieran gestionar varias redes privadas con un sólo gestor. Hasta el momento se ha probado una sola red privada y, como se comenta, el resultado ha sido satisfactorio. En el caso en el que se quieran gestionar más de una red lo que hay que tener en cuenta es que los dispositivos no pueden tener la misma dirección privada, si no la herramienta de gestión no sabrá a qué dispositivo se está refiriendo. Pero, teniendo esto en consideración, y evitando repetir direcciones este problema queda solucionado.

Capítulo 7

Conclusiones y líneas futuras

7.1 CONCLUSIONES

Tras la realización del proyecto se ha llegado a la afirmación de que la elección del protocolo SNMP es óptimo como línea de actuación a seguir ya que:

- Net-SNMP proporciona mecanismos para la extensión de nuevas MIB, así como distintas formas de hacerlo, lo que lo hace vérsatil y adaptable a las necesidades de gestión que se buscan.
- La extensión realizada consume una cantidad razonable de recursos, lo que lo hace óptimo para el caso de estudio en el que la cantidad de recursos es un punto crucial.
- SNMP permite gestionar una amplia gama de dispositivos debido a su universalidad.

Personalmente, este proyecto ha sido muy beneficioso tanto a nivel humano como académico. En lo relativo a mi formación como ingeniera he aprendido, por un lado, un nuevo SO, Linux, el cuál soy capaz de utilizar con soltura. Por otro lado he aprendido, de manera general, todo lo relativo a la gestión de red, protocolo SNMP, herramientas de gestión (Zabbix, Centreon), y de manera particular, cómo actúa el agente dentro de la gestión y cómo extenderlo. He llegado a un buen nivel de programación en C. Así como a utilizar herramientas de depuración como *ddd* [38], *gbd* [39] y *valgrind* [40]. He aprendido más a fondo sobre la arquitectura de un ordenador, cómo hace las llamadas a los sistemas, cómo almacena los resultados, dónde indica los fallos, etc. También he implementado una red real, la red de

prueba5.1, la cuál he tenido que configurar para que funcionase. En general, todos los pasos de un proyecto, desde el análisis de los problemas, tanto técnicos como sociales, hasta buscar la solución, implementarla y probarla. Para complementar todo el proyecto también he tenido que aprender una serie de herramientas ofimáticas como la herramienta gráfica *Dia*, el editor de imágenes libre *GIMP* [41], además del uso de *LaTex* [42] para la presenta memoria.

7.2 EVALUACIÓN DEL CUMPLIMIENTO DE LOS OBJETIVOS

Al inicio del proyecto el objetivo general era mejorar el SGR existente, para ello, se definieron los siguientes objetivos específicos:

1. Estudio del SGR existente enumerando sus fallos, carencias y virtudes para poder definir un nuevo SGR a partir de estas conclusiones. A partir de la definición de este nuevo SGR, el cuál se concluyó que iba a basarse en el protocolo SNMP, se definieron los siguientes objetivos adicionales:

2. Extensión del agente de gestión SNMP implementando una MIB Wifi libre.

3. Estudio de diversas plataformas de gestión de red libres para poder elegir la más adecuada a las necesidades de los usuarios.

4. Estudio de las VPN para poder gestionar la red de estudio desde un punto exterior a la misma accediendo por medio de Internet.

De manera general, puede decirse que se han cumplido con éxito tanto el objetivo 1 como el 2. No obstante, el objetivo 3 y 4, aunque trabajados, no han sido completados del todo.

La causa principal ha sido que la implementación del programa para la extensión del agente ha dado bastantes más problemas de los que se esperaban. Por un lado, se tenía que entender muy claramente cómo extender el agente, y la documentación existente era muy general y escueta por lo que ha sido complicado llegar a comprender exactamente cómo actuaba el agente, qué funciones había que utilizar, qué hacían estas funciones, cómo complementarlas, en general, cómo hacerlo. Ha sido necesario buscar y leer mucha documentación, probar

y, sobre todo, ir depurando para llegar a entenderlo. Por otro lado, también se tenía que tener claro cómo se obtenían los valores que se necesitaban implementar. En este paso, de nuevo, aunque más documentado, se han tenido que implementar diferentes y pequeños programas para entender cómo el sistema obtiene los parámetros, qué parámetros obtiene exactamente, en qué unidades, dónde los almacena, cómo los almacena, etc.

Además, una vez entendido todo, a la hora de fusionarlo para crear el programa final, hubo problemas de programación, los cuales, suelen llevar su tiempo en solucionar, sobre todo los relativos a punteros. Por lo que también hubo que estudiar diferentes maneras de depurar.

Aún así, se han llegado a conclusiones importantes sobre los objetivos que no han llegado a completarse del todo. Respecto a las VPN, se ha demostrado, mediante un escenario de prueba, que se puede crear una VPN permitiendo la gestión de los dispositivos internos por parte del nodo gestor externo. En cuanto a la herramienta de gestión, aunque al principio se invirtieron un par de semanas buscando y comparando distintas herramientas, no se llegó a ninguna conclusión específica. No obstante, cuando otro compañero apostó por Centreon y se procedió a su instalación para probarla, se diseñó y montó la red de pruebas, además de ayudar a terminar de instalar y configurar en Centreon lo que tenía que ver con SNMP para comprobar que, efectivamente, se gestionaba con este protocolo y ya no sólo los parámetros que pueden gestionarse por defecto sino también los parámetros de las MIB adicionales que se habían implementado, integrándose correctamente con la herramienta de gestión.

7.3 POSIBLES MEJORAS Y LÍNEA FUTURA

En primer lugar, hay que ahondar en los puntos que no han sido profundizados del todo. Así pues, por un lado, en lo relativo a la VPN, debe terminar de estudiarse bien la herramienta OpenVPN e implementarla y probarla con la nueva herramienta de gestión Centreon.

Por otro lado, sobre todo, el trabajo principal en la continuación del SGR tiene que ver con la *herramienta de gestión*. Aunque ya está instalada, debe probarse más a fondo, para poder sacar de ella el máximo partido. Hasta ahora se ha configurado para realizar las siguientes funciones:

- Añadir dispositivos para poder ser gestionados especificando qué parámetros gestionar

en ellos y cuál es su situación dentro de la red.

- Obtener los valores que se le han especificado mostrándolos en gráficas en función del tiempo.

- Mostrar un esquema de los dispositivos de la red especificando si estos están o no disponibles, según se muestren en verde o rojo,respectivamente.

Esto es algo realemente útil para tener una primera idea sobre cómo está la red, no obstante, deberían probarse otros aspectos como:

- Autodescubrimiento de dispositivos. Esta función permite descubrir dispositivos nuevos de la red según las caráctarísticas especificadas. Por ejemplo, descubre todos los dispositivos a los que se les pueda hacer ping o todos los que puedan ser gestionados por el protocolo SNMP, o cualquier otra regla. Esto es realmente útil para no tener que añadir manualmente los nuevos dispositivos así como para saber cuáles son todos los dispositivos existente.

- Funcionamiento de las alarmas. Especificar cuándo deberían saltar, así como a quién deberían avisar.

- Creación de diferentes tipo de gráficas, según diferentes criterios.

- Integración con un sistema de gestión de incidencias. Gracias a esta herramienta puede tenerse un historial sobre el tipo de incidencias que ha habido, cómo se solucionaron, quién las solucionó, cada cuánto ocurren, etc. Se está trabajando actualmente en el sistema RT [43] ya que puede integrarse con Centreon.

En general, sacarle más partido a todas las funciones que pueda ofrecer esta herramienta.

Otra mejora sería la relativa a la extensión del agente. Podrían obtenerse más parámetros WiFi que puedan interesar. Se han programado los básicos pero pueden programarse otros muchos. No obstante, completar el programa ya no debería suponer tanto problema. Se ha dejado todo documentado y bastante estructurado para que la ampliación del mismo no suponga demasiado esfuerzo.

7.3. POSIBLES MEJORAS Y LÍNEA FUTURA

Como último punto destacar la importancia de la capacitación de los usuarios finales, los cuales, al fin y al cabo, son los que se encargan de utilizar estas herramientas para conseguir la buscada y deseada sostenibilidad de la red.

Bibliografía

[1] Página principal del programa EHAS. http://www.ehas.org/.

[2] IEEE. *IEEE 802.11-1999 Standard: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Computer Society, Sep 1999.

[3] J.D. Case, M. Fedor, M.L. Schoffstall, and J. Davin. Simple Network Management Protocol (SNMP). RFC 1157 (Historic), May 1990.

[4] Página principal del paquete Net-SNMP. http://www.net-snmp.org/.

[5] Alexander Clemm. *Network Management Fundamentals*. Cisco Press, 2006.

[6] William Stallings. *SNMP,SNMPV2,Snmpv3,and RMON 1 and 2*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1998.

[7] Oscar Ramos Moreno. Sistema de monitorización en diferido basado en correo electrónico, 2005.

[8] Eva Juliana Maya Ortiz. Manual de usuario del sistema de gestión de redes ehas, 2006.

[9] Matthew S Gast. *802.11 Wireless Networks: The Definitive Guide, Second Edition*. O'Reilly Media, Inc., 2005.

[10] Conceptos básicos de tecnología wifi. http://www.virusprot.com/cursos/Redes-Inalámbricas-Curso-gratis.htm.

[11] IEEE. *Revision IEEE 802.11-1999 Standard: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Computer Society, Sep 2007.

[12] IEEE. *Amendment to IEEE 802.11-2007 Standard: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Radio Resource Measurement of Wireless LANs*. IEEE Computer Society, Jun 2008.

[13] IEEE. *Amendment to IEEE 802.11-2007 as amended by IEEE Sdt 802.11k-2008 Standard: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Fast Basic Service Set (BSS) Transition*. IEEE Computer Society, Jul 2008.

[14] IEEE. *Amendment to IEEE 802.11-2007 as amended by IEEE Sdt 802.11k-2008 and Std 802.11r-2008 Standard: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: 3650-3700 MHz*. IEEE Computer Society, Nov 2008.

[15] Página principal del SO Linux. http://www.linux.org/.

[16] Página principal del fabricante Atheros. http://www.atheros.com/.

[17] R.A. Andrade, P. Hernán Salas, and D. Santos Paredes. Tecnología wifi. http://www.cnc.gov.ar/pdf/N5_WI-FI.pdf, 2008.

[18] John Larmouth. *ASN.1 complete*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

[19] Página man de mbrowse del SO.

[20] Página principal de mib2c accesible en la dirección. http://net-snmp.sourceforge.net/docs/man/mib2c.html.

[21] IOCTL página de wikipedia accesible en la dirección. http://es.wikipedia.org/wiki/Ioctl.

[22] Página man de ioctl SO.

[23] Jean Tourrilhes. Wireless tools for linux. http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html#docu, 2008.

[24] Documentación accesible en formato html en la dirección. http://www.net-snmp.org/wiki/index.php/FAQ:Agent_06, Dec 2006.

[25] Helbert Schildt. *C: Manual de referencia*. Osborne-McGraw-Hill, 1995.

[26] Página principal del fabricante WRAP accesible en la dirección. http://www.wrap.org.uk/.

[27] Página principal del fabricante Linksys accesible en la dirección. http://www.linksys.com.

[28] Página principal de la herramienta de gestión Centreon accesible en la dirección. urlhttp://www.centreon.com/.

[29] Página principal de la herramienta de gestión Nagios accesible en la dirección. urlhttp://www.nagios.org/.

[30] Pablo Osuna. Gestión de red y de incidencias. http://wiki.ehas.org/index.php?title=Gestión_de_red_y_de_incidencias, Jun 2009.

[31] K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, and G. Zorn. Point-to-Point Tunneling Protocol (PPTP). RFC 2637 (Informational), July 1999.

[32] C. Madson and R. Glenn. The Use of HMAC-SHA-1-96 within ESP and AH. RFC 2404 (Proposed Standard), November 1998.

[33] Página principal de OpenVPN accesible en la dirección. urlhttp://openvpn.net/.

[34] W. Simpson. The Point-to-Point Protocol (PPP). RFC 1661 (Standard), July 1994. Updated by RFC 2153.

[35] Página principal del fabricante Mikrotik accesible en la dirección. http://www.mikrotik.com/.

[36] Página principal de la herramienta de gestión Zabbix accesible en la dirección. urlhttp://www.zabbix.com/.

[37] Página principal del fabricante Meshlium accesible en la dirección. http://www.meshlium.com/.

[38] Página principal del depurador ddd accesible en la dirección. http://www.gnu.org/software/ddd/.

[39] Página principal del depurador gdb accesible en la dirección. urlhttp://www.gnu.org/software/gdb/.

[40] Página principal del depurador Valgrind accesible en la dirección. http://valgrind.org.

[41] Página principal en español del editor de imágenes libre GIMP accesible en la dirección. urlhttp://www.gimp.org.es/.

[42] Manual de LaTex accesible en la dirección. urltobi.oetiker.ch/lshort/lshort.pdf.

[43] Página principal del SGI RT accesible en la dirección. http://bestpractical.com/rt/.

[44] Nosotros los pueblos: la función de las naciones unidas en el siglo XXI. http://www.un.org/spanish/milenio/index.html, 2000.

[45] A. Martínez, V. Villarroel, J. Seoane, A. Sánchez, and F. del Pozo. Sistemas de telemedicina rural para países en desarrollo. In *Actas CASEIB 2002 del XX Congreso Anual de la Sociedad Española de Ingeniería Biomédica*, pages 395–398, Zaragoza, November 2002.

[46] Andrés Martínez. *Evaluación de impacto del uso de tecnologías apropiadas de comunicación para el personal sanitario rural de países en desarrollo*. PhD thesis, Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad Politécnica de Madrid, 2003.

[47] A. Martínez, V. Villarroel, J. Seoane, and F. del Pozo. Analysis of information and communication needs in rural primary healthcare in developing countries. *IEEE Transactions on Information Technology in Biomedicine*, 9(1):66–72, March 2005.

[48] J. Simó. *Modelado y optimización de IEEE 802.11 para su aplicación en el despliegue de redes extensas en zonas rurales aisladas de países en desarrollo*. PhD thesis, ETSI Telecomunicación, Universidad Politecnica de Madrid, Ciudad Universitaria s/n, 28040 Madrid, Spain, 2007.

[49] Grupo de Telecomunicaciones Rurales. *Redes inalámbricas para zonas rurales*. Pontificia Universidad Católica del Perú, Jan 2008.

[50] Documentación sobre la configuración de PPTP accesible en la dirección. http://www.mikrotik.com/documentation/manual_2.7/Interface/PPTP.html.

Apéndice A

Fundación EHAS

Tres de los ocho Objetivos de Desarrollo del Milenio de Naciones Unidas [44] están relacionados directamente con la salud:

- Objetivo 4: Reducir la mortalidad infantil en dos terceras partes en niños menores de 5 años.
- Objetivo 5: Reducir la mortalidad materna en tres cuartas partes.
- Objetivo 6: Combatir el VIH/SIDA, el paludismo y otras enfermedades.

Cada persona, cada institución, cada país debe identificar aquellos objetivos sobre los que puede influir positivamente y llevarlo a cabo.

Centrándose en países en vías de desarrollo, se consideran las redes inalámbricas en zonas rurales de estos países un medio para mejorar los procesos de atención de salud. Con estas premisas, el programa EHAS, ha trabajado desde 1996 en varías líneas de investigación que persiguen determinar cuáles son las tecnologías inalámbricas más apropiadas en estas zonas, mejorarlas y aplicarlas de forma óptima para mejorar los procesos de salud.

A.1 SISTEMAS DE ATENCIÓN DE SALUD PRIMARIA EN ZONAS RURALES

De forma genérica, los establecimientos de atención primaria en los países en desarrollo se pueden agrupar en dos categorías: Centros de Salud (CS), también llamados en otros

países policlínicos) y Puestos de Salud (también llamados consultorios) [45].

Los PS son los establecimientos de menor jerarquía dentro del sistema público de atención primaria y constituyen la puerta de acceso al sistema para la población rural.

Suelen estar situados en poblaciones de no más de mil habitantes, sin líneas telefónicas y mal dotadas de infraestructura de carreteras.

La mayoría de estos PS están dirigidos por técnicos de enfermería, enfermeras o a lo sumo un médico recién graduado, personal con escasa formación y que necesita comunicación con su médico de referencia para realizar consultas, enviar informes de vigilancia epidemiológica, abastecerse de medicamentos e informar sobre la existencia de brotes epidémicos agudos, emergencias médicas o desastres naturales.

Normalmente, la comunicación e intercambio de información se realiza por desplazamiento del personal de salud entre establecimientos, caminando o en vehículo terrestre o fluvial, lo que puede llevar horas e incluso días.

Varios PS dependen de un único CS, conformando lo que llamaremos *microrred de salud*, que se convierte en la unidad básica de atención primaria. Las microrredes están dirigidas por un médico que es el responsable del CS y que coordina las acciones de los PS que dependen de él.

Los CS son establecimientos de mayor jerarquía que los PS, situados en capitales de provincia o distrito, donde suele llegar la línea telefónica. Un CS es centro de referencia de varios PS. Está siempre dirigido por médicos y posee cierta infraestructura y equipamiento para realizar algunas pruebas diagnósticas, además de permitir a veces hospitalización.

Son el lugar desde el que se coordinan las actividades de los PS asociados.

A.2 JUSTIFICACIÓN DE REDES INALÁMBRICAS EN ZONAS RURALES

Al llevar a cabo un estudio sobre las necesidades de los trabajadores de los centros y puestos de salud de estas zonas se vio que:

- Las ¾ partes del personal sanitario rural tienen sensación de aislamiento profesional.

- La mayoría de los establecimientos de salud rural están dirigidos por técnicos de enfermería, personal con escasa formación que necesita comunicación continua con su médico de referencia para hacer consultas clínicas.
- Entre 1 y 2 días a la semana quedan desatendidos los establecimientos rurales por viajes de coordinación del personal asistencial.
- La media de tiempo necesaria para que un técnico viaje hasta su centro de referencia (lugar donde se encuentra a su médico responsable) es muy alta (por ejemplo, en la provincia del Alto Amazonas de Perú es de 10 horas ida y 10 horas vuelta).
- Hay un gasto por el envío de información epidemiológica y administrativa (los puestos de salud envían alrededor de 100 hojas mensuales a su centro de salud de referencia, y los centros de salud unas 300 a la dirección provincial).
- El personal sanitario es muy joven (alrededor de 32 años) y existe una alta rotación (no permanecen más de 2 años en el mismo establecimiento).
- La mayoría de los establecimientos de salud rurales no tiene posibilidad de instalar teléfono, ni está en los planes a medio plazo de las compañías telefónicas.
- No hay acceso a electricidad en la mayoría de las poblaciones rurales.

Tras un largo estudio, investigación e instalación de redes inalámbricas y, transcurrido un cierto tiempo de funcionamiento efectivo de las mismas se pudo afirmar que, se habían mejorado en gran medida los siguientes puntos esenciales, mejorando los procesos de atención de salud [46]:

- Formación a distancia de médicos, enfermeras y técnicos de enfermería.
- Automatización del sistema de vigilancia epidemiológica. Desarrollando un sistema informatizado de vigilancia epidemiológica capaz de cubrir las necesidades de recogida, envío, procesado, visualización y realimentación de información a nivel nacional.
- Consultas remotas y referencias de pacientes a niveles jerárquicos superiores.
- Mejora de los sistemas de evacuación de emergencias y distribución de medicamentos.

Quedando patente la importancia de las redes inalámbricas como forma de mejorar los procesos de salud.

A.3 ALTERNATIVAS TECNOLÓGICAS

Como se viene diciendo es imprescindible que las soluciones tecnológicas se adapten al entorno en el cuál se van a utilizar por lo que deben cumplir una serie de características [47]:

- Tiene que ser robusta y sencilla de usar, ya que los usuarios van a ser poco cualificados y no van a contar con el apoyo continuado de asesores preparados.
- Tiene que requerir poco o ningún mantenimiento de técnicos especializados, ya que éstos van a estar lejos y va a resultar caro y difícil atraerlos para la resolución de los problemas. Con más razón debe ser mínima la necesidad de administración de las redes, ya que ésta genera costes fijos considerables.
- Debe ser de bajo consumo, ya que frecuentemente tendrá que depender de instalaciones de energías fotovoltaicas o eólicas que encarecen las instalaciones y aumentan las necesidades y costes de mantenimiento.
- Debe tener costes de despliegue y de operación muy bajos. Ésto excluye las redes cableadas, las de telefonía móvil y las redes satélite como soluciones únicas. En ocasiones se puede plantear el acceso al mundo de toda una red por estos medios, pero la distribución del acceso de tendrá que hacer con una tecnología complementaria más barata. Este criterio también desaconseja en muchos casos las redes radio en bandas de frecuencia licenciadas.

Teniendo en cuenta estos requerimientos se irán explicando las diferentes tecnologías usadas, aplicadas o en fase de estudio.

A.3.1 WIFI

La familia de estándares IEEE 802.11, más conocida como WiFi, tiene asignadas las bandas ISM *(Industrial, Scientific and Medical)* 902-928 MHz, 2.400-2.4835 GHz, 5.725-

5.850 GHz con objeto de lograr redes de área local inalámbricas WLAN *(Wireless Local Area Network)*.

Aunque este estándar no se concibió para crear redes extensas, sus indudables ventajas de costo, uso de frecuencias libres de licencia y gran ancho de banda, han despertado el interés de diversos agentes tecnológicos de países en desarrollo. Además, el enorme éxito de WiFi en todos los ámbitos ha dado lugar a una gran cantidad de productos en el mercado, casi todos ellos de bajo consumo, a precios bajos y mucha flexibilidad de uso, especialmente en combinación con desarrollos de SW libre.

Por lo tanto y dado que la tecnología WiFi fue en su inicio diseñada para redes locales, la mayor dificultad ha residido en su aplicación para largas distancias.

Problemática del uso de WiFi para largas distancias

WiFi comparte la mayoría de su funcionamiento interno con Ethernet, sin embargo difiere en la especificación de su capa física (PHY) utilizando señales radio en lugar de cable y en su capa de acceso al medio (MAC). A continuación se detallan los aspectos a tener en cuenta en cada capa para poder utilizar el estándar para largas distancias.

Capa física Una cuidadosa revisión del estándar no deja entrever ningún elemento de la capa física que limite el alcance de las comunicaciones. Los límites físicos dependerán, por lo tanto, de los siguientes parámetros:

- Máxima potencia transmitible PIRE
- Pérdidas de propagación. Este parámetro dependerá, sobre todo, de los fenómenos meteorológicos, del tipo de polarización y de las interferencias, aunque este último parámetro no afectará en zonas rurales aisladas. En cuanto el tipo de polarización destacar que las condiciones atmosféricas y el terreno pueden producir una cierta despolarización, empeorando la recepción de la señal y aumentando su atenuación. El mejor comportamiento se da con polarización vertical.
- Sensibilidad de recepción. Los diferentes modos del estándar usan diferentes tipos de modulación y codificación de modo que, cuanto mayor sea la velocidad, mayor es la potencia necesaria en recepción para mantener un enlace con una BER baja. Esta

potencia, que es la llamada sensibilidad, obliga a usar velocidades bajas si se quiere lograr enlaces de larga distancia con una cierta estabilidad.

- Mínima relación señal a ruido que se esté dispuesto a aceptar como suficiente.

Capa MAC La capa MAC tiene multitud de tiempos constantes definidos que tienen diferente efecto en función de la distancia que haya entre las estaciones. Tras una revisión cuidadosa del estándar realizada por Francisco Javier Simó [48], se pueden extraer tres tipos de limitaciones: el temporizador de espera de los ACK's, la definición de los tiempos relacionados con el SlotTime, y el cálculo del vector que se encarga de controlar el tiempo que se debe esperar cuando el canal está reservado para la detección de la portadora virtual NAV *(Network Allocation Vector)*.

- ACKTimeout. Este parámetro, que se corresponde con el tiempo en el que la estación transmisora espera la llegada de un ACK una vez que se ha terminado la transmisión de un paquete, toma unos ciertos valores dentro de un rango teniendo en cuenta el tiempo de ida y vuelta, por la tanto, la distancia. Esto quiere decir que si las estaciones receptora y transmisora están demasiado alejadas este tiempo pasará antes de que pueda llegar el ACK a la estación transmisora, pensando ésta que ha habido un fallo, por lo que procederá a la retransmisión del paquete.

- Slottime. El estándar prevé que las estaciones que transmiten son oídas por las otras estaciones dentro del mismo *slot* en que se ha producido la transmisión, lo cuál impone un límite de unos 3Km. Más allá de esa distancia, las prestaciones de los enlaces empeoran con la distancia, aunque aún resultan utilizables si el número de nodos activos es suficientemente bajo.

- La vulnerabilidad con nodos ocultos. En IEEE 802.11 se emplea el mecanismo RTS/CTS para evitar colisiones entre nodos ocultos. No obstante, ese mecanismo funciona si el cómputo del NAV se corresponde con el tiempo que verdaderamente el canal va a permanecer ocupado. Puesto que el NAV no se calcula teniendo en cuenta el tiempo de propagación, a medida que la distancia aumenta su efectividad empeora. En enlaces punto-multipunto con distancias del orden de kilómetros, el RTS/CTS es prácticamente inservible, y no hay un mecanismo alternativo.

Como consecuencia de lo anterior y, dependiendo del tipo de enlace que define la arquitectura de red 802.11, se pueden sacar las siguientes conclusiones:

- Punto-Punto. Cuando la distancia es mayor de 3Km, se incrementa proporcionalmente con la distancia, en saltos de 3Km, el número de *slots* en que una estación puede empezar a transmitir y colisionar con un paquete cuya transmisión se inició en un *slot* determinado. Esto tiene relativamente poco impacto cuando la carga ofrecida es baja, pero es importante cuando el enlace está próximo a la saturación, ya que en ese caso casi siempre hay un paquete listo para ser transmitido tan pronto como se considere libre el canal, y para ventanas de contención pequeñas la probabilidad de colisión será significativa. También será necesario cuidar el ajuste del ACKTimeout fijándolo a un valor ligeramente superior a dos veces el tiempo de propagación.

- Punto-Multipunto. Además de darse las mismas anomalías de comportamiento que las comentadas para Punto-Punto, hay que tener en cuenta las otras estaciones que observan pasivamente el canal esperando que se desocupe, ya que tomarán decisiones equivocadas al considerar el canal libre cuando no lo está.

Según lo expuesto anteriormente, puede concluirse que WiFi puede servir, aunque con cierta pérdida de prestaciones, para enlaces punto-punto de larga distancia si los equipos terminales permiten configurar el ACKTimeout y el SlotTime, en cambio, para punto-multipunto, aún modificando esos parámetros, el funcionamiento es notablemente peor a menos que la carga ofrecida y el número de nodos sean muy bajos.

Actualmente esta tecnología es la opción elegida para implementar redes inalámbricas. El uso de WiFi permite unos anchos de banda capaces de ofrecer una gran cantidad de servicios (VoIP, navegación web y correo electrónico). Además es un estándar ampliamente conocido y fácil de configurar. Ofrece una gran flexibilidad a la hora de adherir un nuevo nodo y el consumo de potencia es bajo, menor de 10 W por router. A pesar de que requiere línea de vista directa y que necesite repetidores para solventar los problemas de distancias de decenas de kilómetros, es la tecnología que más ventajas ofrece.

A.3.2 VHF

Las redes privadas de comunicación de voz en banda VHF utilizan la banda de frecuencia de 30-300 MHz. En ellos se alcanzan distancias de enlace en torno a los 70 Km, limitados por la potencia de transmisión y la altura de las antenas. Éstas deberán compensar la curvatura de la tierra y salvar los obstáculos que se presentan en el camino, aunque tiene bastante tolerancia a los mismos.

En la propagación directa desde la antena transmisora a la antena receptora es recomendable que exista línea de vista entre ellas, es decir, que exista visibilidad óptica entre ambas. Sin embargo, se soportan obstáculos vegetales o invasiones no muy profundas de la línea de vista por elevaciones del terreno.

El inconveniente de no lograr un enlace debido a obstrucción severa de la línea de vista puede superarse utilizando equipos intermedios o repetidores, usualmente ubicados en zonas elevadas, de forma que permitan la comunicación, a través de ellos, entre dos o más puntos que no tienen visibilidad directa.

Aunque está pensada para comunicación de voz, mediante SW se consigue también la comunicación de datos, aunque la velocidad que se consigue es muy baja.

Esta tecnología fue una primera opción óptima para enlaces a largas distancias y fue la usada hasta que WiFi se mostró como la opción que más ventajas ofrecía.

A.3.3 HF

La transmisión en banda HF tiene un rango de frecuencia de operación de 3-30 MHz y permite comunicaciones de centenares y hasta miles de kilómetros. El método de propagación para la banda HF es el conocido por *onda ionosférica*, por el cual las ondas de transmiten y luego se reflejan hasta el punto de recepción.

La propagación depende de las capas de la ionosfera, que poseen un comportamiento aleatorio en cuando a estructura y densidad, afectando directamente a las frecuencias de trabajo. Para poder tener en cuenta estas características y poder configurar los equipos a una frecuencia adecuada para las condiciones climatológica, época, hora y lugar, se determinan frecuencias mínimas LUF y máximas MUF de trabajo. Por debajo de la LUF no se podría realizar propagación por la ionosfera, debido a que hay gran absorción y, a frecuencias ma-

yores de la MUF, las ondas atraviesan la ionosfera y no son reflejadas a la tierra.

Por lo tanto, los sistemas de radio HF habilitan comunicaciones a través de terrenos planos, elevados o montañosos, sin la necesidad de dispositivos de retransmisión, como los repetidores.

Al igual que la banda VHF, HF también está destinada a la transmisión de voz y por sus características no es recomendable para la transmisión de datos ya que los enlaces son de peor calidad y tienen mucha variabilidad. Además el entorno es muy sensible a errores por los desvanecimientos ocasionados por la incertidumbre de la propagación ionosférica.

Por estos motivos, sobre todo va a utilizarse para alcanzar lugares con ubicaciones complicadas sin necesidad de repetidores.

A.3.4 WiMAX

IEEE 802.16, más conocida como WiMAX, es el fruto del trabajo realizado entre 2002 y 2005 en el IEEE para la definición de nuevas propuestas tecnológicas que permiten cubrir las lagunas de las redes inalámbricas de banda ancha. Es decir, posibilitar redes inalámbricas de altas prestaciones en áreas metropolitanas sin línea de vista.

El principal problema deriva de los costos de instalación y de lo nuevo de esta tecnología.

A.3.5 Conexión Satelital (VSAT)

El mercado de las telecomunicaciones satelitales bidireccionales de voz y datos está prácticamente copado en Hispanoamérica por una única tecnología: VSAT *(Very Small Aperture Terminal)*. Las redes VSAT son redes de comunicación de datos vía satélite para el intercambio de información punto-punto o punto-multipunto.

El componente principal de este sistema es el *hub*, que es la estación central terrestre de la red. Éste permite realizar la comunicación entre dos terminales VSAT, es decir, que todo intercambio de información tiene que pasar por el hub. Esta estructura de red logra que las estaciones terminales sean simples, baratas y fáciles de instalar. Las antenas usadas tienen menor diámetro (menores de 2.4m, típicamente 1.3m) y los sistemas un bajo consumo de energía. Con esta tecnología se consiguen diseñar redes muy densas con altas velocidades de transmisión si hay pocos usuarios conectados simultáneamente, permitiendo la transmisión

de voz, datos y vídeo.

No obstante, los costes de instalación y mantenimiento están fuera del alcance de muchos entornos rurales. Además, necesitan un gran subsistema eléctrico para funcionar, muy costoso en zonas donde apenas hay energía eléctrica. Por lo tanto, aunque sea un sistema robusto, de buena calidad, de fácil y rápida instalación, no es una opción óptima para los casos de estudio.

Apéndice B

La red del río Napo

El presente proyecto fin de carrera centra su estudio en la red que la Fundación EHAS tiene desplegada en el río Napo en la provincia de Loreto de Perú.

El esquema de la red puede verse en la Figura B.1

La distribución de esta red así como sus características principales vienen determinadas por las especificaciones del terreno. Así puede verse como el único trayecto posible, el río, a falta de carreteras, determina un único camino lineal para la red.

Debido a esto la tecnología utilizada es Wifi, estándar 802.11g en 2.4GHz de la banda ISM (*Industrial, Scientific and Medical*), punto a punto. El gran inconveniente es que determina una única vía para la información de la red, por lo que cada nodo se convierte en vital para el correcto funcionamiento de todos los servicios ofrecidos.

Como puede observarse, la red está formada por repetidores y estaciones clientes. La estación repetidora se encuentra instalada en la torre mientras que la estación cliente se encuentra instalada en el establecimiento de salud.

Los enlaces que interconectan las estaciones repetidoras formaran la *red troncal* (en azul). Esta red troncal será la encargada de transportar el tráfico de datos generado por las comunicaciones entre las estaciones cliente.

Asimismo en cada poblado existirá un enlace inalámbrico entre la estación repetidora y la estación cliente. Este enlace permitirá a las estaciones clientes acceder a la red troncal. A este enlace se le llama *red de distribución* (en verde) [49].

De los 17 establecimientos que forman la red, 12 se corresponden con puestos de salud (Torres Causana, Tempestad, Angoteros, Campo Serio, Rumi Tumi, San Rafael, Tacsha Cu-

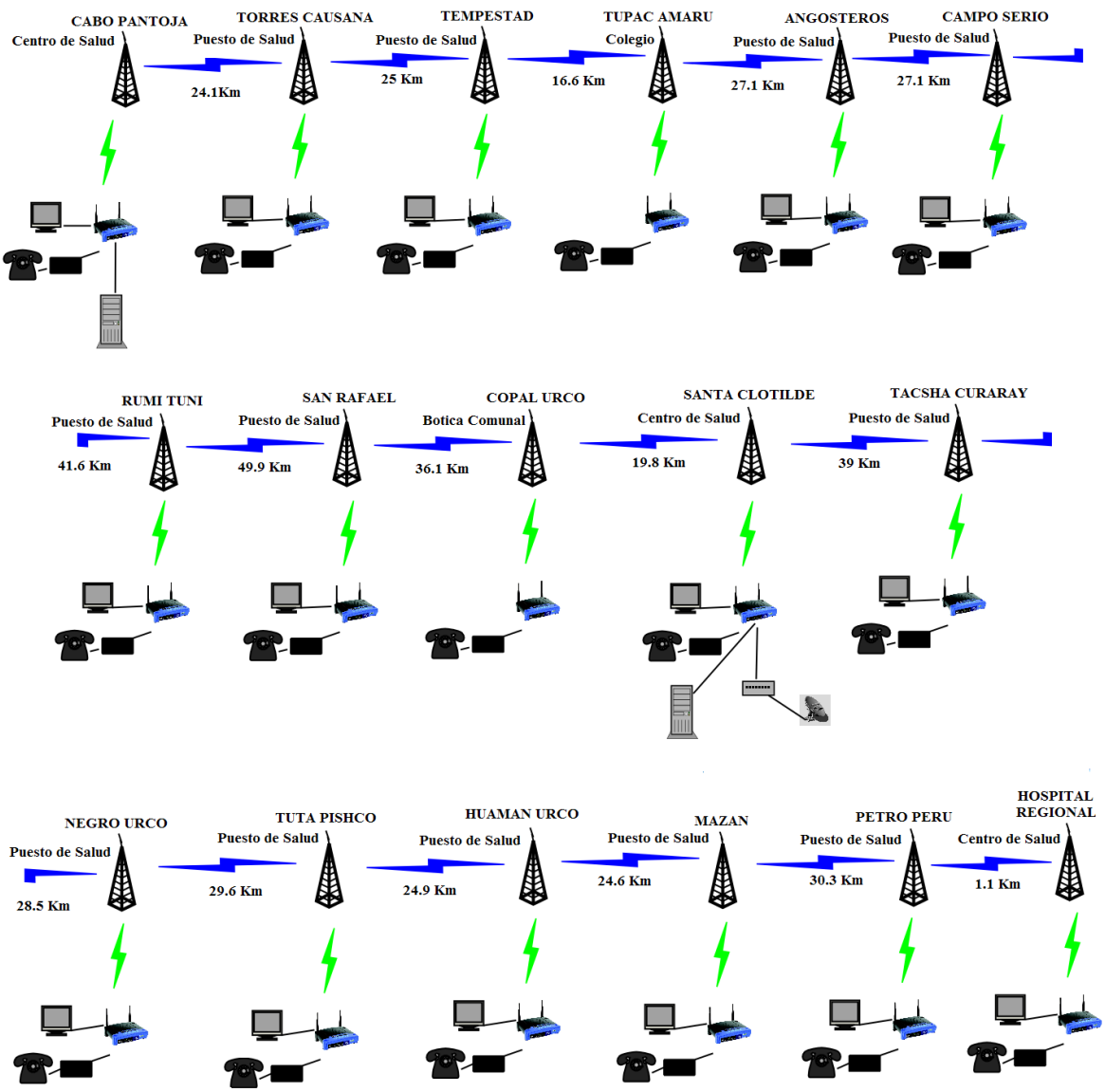

Figura B.1: Esquema de la Red del Río Napo (Perú)

raray, Negro Urco, Tuta Pishco, Huaman Urco, Mazan y Petro Perú), 3 se corresponden con centros de salud (Cabo Pantoja, Santa Clotilde y Hospital Regional), 1 se correponde con un colegio (Tupac Amaru) y otro con una botica comunal (Copal Urco).

Todos estos establecimientos están formados por unos elementos comunes, que son:

- En el establecimiento de salud:
  - El enrutador inalámbrico, en este caso, modelo *LinkSys,* que sirve para enlazar el establecimiento con la red troncal.
  - Un ATA (*Adaptador de Teléfono Analógico*), el cual sirve para convertir las señales entregadas por el teléfono a una forma compatible con una red de datos TCP/IP y viceversa, es decir, se encarga de conectar un teléfono analógico a una red de VoIP.
  - Un teléfono analógico.

  Con estos elementos se aseguran las comunicaciones telefónicas entre los distintos establecimientos.

- En la torre:
  - Un par de routers *WRAP* interconectados por un cable coaxial, cada uno con su tarjeta inalámbrica correspondiente y una tarjeta Compact Flash de memoria, dónde se instala el SO.
  - Antenas directivas de grilla

Además, en los puestos de salud y centros de salud, se encuentran un ordenador y una impresora para el uso de los servicios ofrecidos, correo electrónico y navegación por Internet, así como para el cumplimiento de las diversas labores administrativas que se necesiten realizar.

Por último, destacar que, tanto en Santa Clotilde como en Cabo Pantoja, se encuentran los servidores de correo electrónico y, en Santa Clotilde, además, se encuentra el enlace satelital que es el que proporciona al acceso a Internet a toda la red.

Apéndice C

Implementación de la VPN

Especificar que esta implementación se corresponde con el escenario de la Figura 6.3.

C.1 PPTP

C.1.1 Mikrotik: Servidor

Lo primero que hay que comprobar en el SO del router Mikrotik es que posee el paquete *ppp*. Una vez que se tiene este paquete los pasos a seguir son los siguientes:

1. Añadir un cliente ppp al servidor. Para ello dentro del directorio /ppp se especifica lo siguiente:

   ```
   /ppp secret name="prueba" service=pptp password="prove"
   local-address=10.0.0.100 remote-address=10.0.0.3
   ```

   Por un lado tenemos el nombre y contraseña de usuario, necesarias tanto para la autenticación como para la encriptación. La dirección local, que se refiere a la dirección IP dinámica de la Mikrotik, servidor. La dirección remota, que se corresponde con la dirección IP dinámica del cliente.

2. Añadir ese cliente a la lista del servidor. Para ello dentro del directorio /interface se especifica:

   ```
   /interface pptp-server add name=Llegando user=prueba
   ```

Básicamente se ha enlazado un nombre de cliente pptp con el del cliente ppp recién creado.

3. Habilitar el servidor. Para ello, dentro del directorio /interface se especifica:

```
/interface pptp-server Server set enable=yes
```

4. Habilitar un Proxy-ARP para poder comunicar el cliente pptp con la red privada como si fuesen una única red. No hay que olvidar que dentro de una misma red los dispositivos se comunican mediante el protocolo ARP. Para ello dentro del directorio /interface se especifica:

```
/interface ethernet set ether2 arp=Proxy-arp
```

C.1.2 Cliente

El cliente utilizado ha sido un portátil con un SO basado en Linux, en concreto, un Ubuntu Hardy. Lo primero que se tiene que comprobar es que se ha instalado el paquete *pptp-linux* y *ppp* de los repositorios.

Una vez hecho esto se procede de la siguiente manera:

1. Dentro de la carpeta */etc/ppp* se encuentra un archivo llamado *chap-secrets*. En él se va a guardar el nombre y contraseña de usuario que va a permitir la autenticación y el encriptado. Por lo tanto, teniendo en cuenta lo especificado en el servidor, hay que añadir la siguiente línea:

```
prueba mikrotik prove *
```

Siendo *prueba* y *prove* el nombre y contraseña, respectivamente, del cliente ppp que se ha creado en el servidor pptp. *Mikrotik* es un archivo de configuración del túnel que está situado en */etc/ppp/peers* y que ahora se explicará.

2. Hay que crear el archivo de configuración del túnel. El cuál, como se ha comentado, se encuentra en */etc/ppp/peers* y es el siguiente:

```
pty "pptp 138.4.10.71 --nolaunchpppd"
lock
noauth
```

```
nobsdcomp
nodeflate
name prueba
remotename mikrotik
ipparam mikrotik
debug
```

Una vez que se encuentra todo configurado, para crear el túnel se ejecuta desde el cliente el siguiente comando:

```
sudo pon mikrotik
```

Mientras que, para eliminar el túnel, se ejecuta desde el cliente el siguiente comando:

```
sudo poff mikrotik
```

Como última reseña, destacar que, si se quiere acceder a la red privada, hay que añadir la ruta en el cliente. Para ello se ejecuta:

```
sudo route add -net 10.0.0.0/8 dev ppp0
```

Siendo 10.0.0.0/8 la red privada.

C.2 OpenVPN

Antes de empezar, destacar que, para garantizar la seguridad, se van a crear unos certificados en el cliente, a continuación se explicará la forma de crearlos, los cuales van a tener que se transmitidos de forma segura al servidor, bien sea por ssh, scp, etc, situándolos en la carpeta */certificate import* del servidor.

C.2.1 Mikrotik: Servidor

1. Definir un rango de IP, también denominado IP pool. Para ello, dentro de la directorio *ip* se especifica:

   ```
   /ip pool add name=ovpn-pool ranges=10.0.0.101-10.0.0.200
   ```

2. Definir un perfil ppp. Para ello dentro del directorio *ppp* se especifica:

```
/ppp profile add change-tcp-mss=default comment=""
local-address=10.0.0.4, name=opentunel only-one=default
remote-address=ovpn-pool, use-compression=default,
use-encryption=required use-vj-compression=default
```

3. Definir un usuario vpn. Para ello, dentro del directorio *ppp* se especifica:

```
/ppp secret add caller-id="" comment="" disabled=no
limit-bytes-in=0, limit-bytes-out=0 name="prueba2",
password="prove" routes="" service=any profile=opentunel
```

4. Configurar el servidor OpenVPN. Para ello, dentro del directorio */interface* se especifica:

```
/interface ovpn-server server set auth=sha1,
md5 certificate="el que se ha creado", cipher=blowfish128,
aes128,aes192,aes256 default-profile=opentunel enabled=yes
keepalive-timeout=disabled max-mtu=1500 mode=ip netmask=29 port=1194
require-client-certificate=no
```

5. Habilitar el acceso a OpenVPN por el puerto 1194. Para ello, desde el directorio *ip* se especifica:

```
/ip firewall filter add action=accept chain=input
comment="OpenVPN", disabled=no, dst-port=1194 protocol=tcp
```

C.2.2 Cliente

En primer lugar hay que instalar los paquetes *openvpn* y *openssl*.

Una vez hecho esto, se procede a la creación de los certificados. Para ello, en primer lugar hay que copiar la carpeta */usr/share/doc/openvpn/examples/easy-rsa* a */etc/openvpn*.

Una vez copiada, se modifican los parámetros *KEYCOUNTRY, KEYPROVINCE, KEYCITY, KEYORG,KEYMAIL* del archivo */etc/openvpn/easy-rsa/2.0/vars*.

Una vez hecho esto, se ejecutarán los siguientes comandos como *root*.

1. source ./vars

2. ./clean-all Con este comando se eliminan las claves que pudiesen estar hechas.

3. ./build-ca Este comando crea los certificados de la autoridad. De todos los campos que va a pedir mientras se crea el más importante es el *common name*.

4. ./build-key-server server Con este comando se van a crear las claves del servidor. De nuevo, pedirá rellenar una serie de campos y, de nuevo, el más importante, se corresponderá con *common name*, el cual se refiere al servidor.

5. ./build-key cliente Con este comando se van a crear las claves del cliente. Una vez más, pedirá rellenar una serie de campos y, una vez más, el más importante, se corresponderá con *common name*, el cual se refiere al cliente.

6. ./build-dh Este comando crea los certificados Diffie-Hellman.

Una vez creados todos los certificados, los cuales se encuentran en */etc/openvpn/easy-rsa/2.0/keys*, como ya se comentó, hay que enviarlos al servidor. En concreto hay que enviar al servidor *ca.crt, ca.key, server.key, server.crt, dh1024.pem*.

Por su lado, en la parte del cliente, hay que copiar en */etc/openvpn*, *ca.crt, cliente.key, cliente.crt*.

Una vez hecho esto se creará un fichero de configuración, el cual se situará en */etc/openvpn* y es el siguiente:

```
#/etc/openvpn/cliente.conf
client
# Dispositivo tunel
dev tun
# proto tcp-client
proto tcp
# Dirección real del servidor
remote 138.4.10.71
port 1194
local 138.4.10.122
# me falta una linea muy importante
tls-client
# para verificar el certificado del servidor al que esta conectado
remote-cert-tls server
# Claves y certificados
```

```
ca ca.crt
cert cliente.crt
key cliente.key
# comp-lzo
# resolv-retry infinite
ping 15
ping-restart 45
ping-timer-rem
persist-key
persist-tun
verb 4
# cipher AES-256-CBC
# auth SHA1
# pull
auth-user-pass auth.cfg
# auth-user-pass
pull
# user nobody
# group nogroup
```

Para lanzarlo hay que ejecutar el siguiente comando:

```
sudo openvpn cliente.conf
```