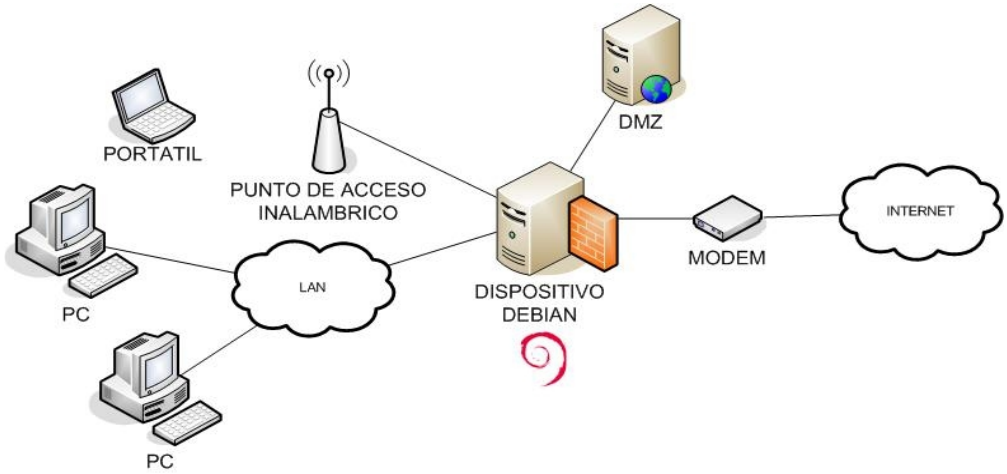
2.2 CONECTIVIDAD DE NUESTRO DISPOSITIVO

2.2.1 Arquitectura de Red de nuestro Sistema

2.2.1.1 Esquema de equipos y conexiones

El sistema informático que vamos a estudiar y en el que vamos a trabajar tiene una arquitectura como la que muestra la siguiente ilustración. El dispositivo que estamos construyendo es el DISPOSITIVO DEBIAN.



En esta ilustración podemos ver cómo nuestro dispositivo actúa de puerta de enlace o gateway para la red local. Todo el tráfico de datos que salga al exterior (Internet) atravesará nuestro dispositivo. Las estaciones de trabajo se encuentran todas en la misma subred interconectadas con cables de pares de cobre que terminan en switches (conmutadores) o hubs, representado en el esquema mediante la nube LAN. En la siguiente sección entraremos en detalle en las características de estos equipos de conmutación.

Nuestro dispositivo tendrá como mínimo dos interfaces de red. Una de ellas será el puerto LAN que actúa como terminador para el cableado de la infraestructura de red local. El otro es el puerto WAN el cual conecta el dispositivo con el exterior: con Internet. Las tecnologías de acceso a Internet son diversas: fibra óptica, cable de pares (adsl), cable coaxial, medio inalámbrico... y diversos protocolos y estándares: PPP, ATM, LMDS, Frame Relay, MPLS, etc... Nuestro dispositivo es independiente de la tecnología de acceso utilizado, ya que es el módem el que se encarga de convertir la tecnología de acceso final en tecnología Ethernet de cable de pares. Nosotros utilizaremos en nuestra batería de pruebas accesos ADSL, por ser accesos de banda ancha muy asequibles para el usuario doméstico y la pequeña empresa.

Adicionalmente el dispositivo puede llevar instalado más interfaces de red o NIC. Podríamos utilizar alguna de ellas como puerto DMZ. DMZ o zona desmilitarizada es aquella zona de nuestra red donde se sitúan los servidores a los que se puede acceder desde el exterior. La DMZ está lógicamente separada de la red local para evitar que si algún atacante remoto toma el control de un servidor, éste no pueda acceder al resto de las estaciones de trabajo de la red. Posteriormente estudiaremos en detalle la zona desmitalizarada y los accesos que son permitidos a ésta.

Por último el punto de acceso inalámbrico hace de conversor del medio cable a aire, permite a los dispositivos móviles y estaciones de trabajo portátiles comunicarse con las estaciones de trabajo de sobremesa y salir a Internet a través de nuestro dispositivo.

2.2.1.2 Hardware utilizado

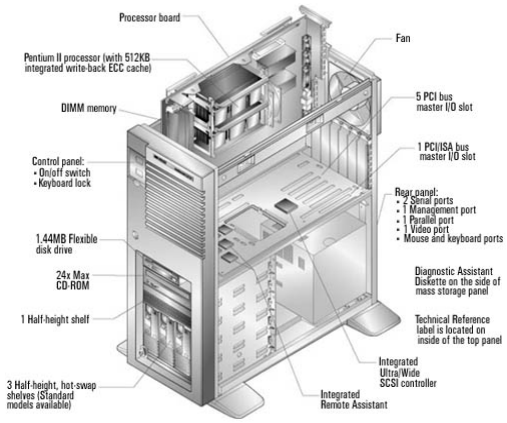
Como comentamos en la introducción, utilizaríamos para este proyecto un equipo antiguo ya que aún así sería suficiente para encaminar paquetes de datos pues este proceso no requiere demasiada carga de CPU. El equipo utilizado para la batería de pruebas y simulación de escenarios reales es un Pentium II a 350MHz con 128MB de memoria RAM. Nuestro equipo dispone de tres interfaces de red, es decir, tres puntos de interconexión con otros equipos de red y/u ordenadores, además de un disco duro de 4.3GB.

¿Cómo ver el Hardware que tenemos en nuestro equipo Debian?

apt-get install lshw
lshw

Lshw es un comando que lista el Hardware encontrado.
He aquí un resumen de lo que muestra por pantalla en nuestra máquina Debian:

```
debian
    description: Computer
    width: 32 bits
  *-core
       description: Motherboard
     *-memory
          size: 128MB
     *-cpu
          product: Pentium II (Deschutes)
          vendor: Intel Corp.
          size: 350MHz
          width: 32 bits
 pse36 mmx fxsr
        *-cache:0
             size: 32KB
        *-cache:1
             size: 512KB
           *-display
                description: VGA compatible controller
                product: 86C326 5598/6326
                size: 8MB
              *-cdrom
                   description: IDE CD-ROM
                   product: TOSHIBA CD-ROM XM-6302B
                   vendor: Toshiba
                   physical id: 1
                   bus info: ide@0.1
                   logical name: /dev/hdb
        *-network:0 DISABLED
             description: Ethernet interface
             product: DECchip 21040 [Tulip]
             vendor: Digital Equipment Corporation
             physical id: 9
             bus info: pci@00:09.0
             logical name: eth0
             serial: 00:80:c8:0c:78:c0
        *-network:1
             description: Ethernet interface
             product: DECchip 21041 [Tulip Pass 3]
             vendor: Digital Equipment Corporation
             physical id: a
             bus info: pci@00:0a.0
             logical name: eth1
             serial: 00:00:1a:00:03:e2
        *-network:2 DISABLED
             description: Ethernet interface
             product: RTL-8029(AS)
             vendor: Realtek Semiconductor Co., Ltd.
             logical name: eth2
             serial: 00:4f:49:04:93:c0
```



2.2.2 Dispositivos físicos de las Redes LAN

2.2.2.1 Introducción a las Redes de Area Local (LAN)

Las redes de datos se desarrollaron como consecuencia de aplicaciones comerciales diseñadas para microcomputadores. Por aquel entonces los microcomputadores no estaban conectados entre sí como sí lo estaban las servidores de terminales *mainframe,* por lo cual no había una manera eficaz de compartir datos entre varios ordenadores. Se tornó evidente que el uso de disquetes para compartir datos no era un método eficaz ni económico para desarrollar la actividad empresarial.

Las empresas necesitaban una solución que resolviera con éxito los tres problemas siguientes: Cómo evitar la duplicación de equipos informáticos y de otros recursos. Cómo comunicarse con eficiencia. Cómo configurar y administrar una red. Las empresas se dieron cuenta de que la tecnología de **networking** podía aumentar la productividad y ahorrar gastos. Las redes se agrandaron y extendieron casi con la misma rapidez con las que se lanzaban nuevas tecnologías y productos de red. A principios de la década de 1980 las tecnologías de red se expandieron enormemente, aun cuando en sus inicios sus desarrollos fueron desorganizados.

A mediados de la década de los 80, las tecnologías de red que habían emergido se habían creado con implementaciones de hardware y software distintas. Cada empresa dedicada a crear hardware y software para redes utilizaba sus propios estándares corporativos. Estos estándares individuales se desarrollaron como consecuencia de la competencia con otras empresas. Por lo tanto, muchas de las nuevas tecnologías no eran compatibles entre sí. Se tornó cada vez más difícil la comunicación entre redes que usaban distintas especificaciones. Esto a menudo obligaba a deshacerse de los equipos de la antigua red al implementar equipos de red nuevos.

Una de las primeras soluciones fue la creación de los estándares de **Red de área local** (**LAN** - **Local Area Network, en inglés)**. Como los estándares LAN proporcionaban un conjunto abierto de pautas para la creación de hardware y software de red, se podrían compatibilizar los equipos provenientes de diferentes empresas. Esto permitía la estabilidad en la implementación de las LAN.

Las LAN se encuentran diseñadas, por tanto, para:

- Operar en un área geográfica limitada
- Permitir el multiacceso a medios con alto ancho de banda
- Controlar la red de forma privada con administración local
- Proporcionar conectividad continua a los servicios locales
- Conectar dispositivos físicamente adyacentes

En un sistema LAN, cada departamento de la empresa era una especie de isla electrónica. A medida que el uso de los computadores en las empresas aumentaba, pronto resultó obvio que incluso las LAN no eran suficientes. Lo que se necesitaba era una forma de que la información se pudiera transferir rápidamente y con eficiencia, no solamente dentro de una misma empresa sino también de una empresa a otra. La solución fue la creación de redes de área metropolitana (**MAN**) y redes de área amplia (**WAN**). Como las WAN podían conectar redes de usuarios dentro de áreas geográficas extensas, permitieron que las empresas se comunicaran entre sí a través de grandes distancias.

Una de las configuraciones comunes de una LAN es una **red interna**, a veces denominada "intranet". Los servidores Web de red interna son distintos de los servidores Web públicos, ya que es necesario que un usuario público cuente con los correspondientes permisos y contraseñas para acceder a la red interna de una organización. Las redes internas están diseñadas para permitir el acceso por usuarios con privilegios de acceso a la LAN interna de la organización. Dentro de una red interna, los servidores Web se instalan en la red. La tecnología de navegador se utiliza como interfaz común para acceder a la información, por ejemplo datos financieros o datos basados en texto y gráficos que se guardan en esos servidores.

2.2.2.2 Tecnologías y Dispositivos Físicos (Hardware) de Red

Desde el punto de vista físico, (capa 1 del modelo OSI), el hardware más utilizado para las redes de acceso local (LAN) es conocido como Ethernet (o FastEthernet o GigabitEthernet). Sus ventajas son su bajo coste, velocidades aceptables (10, 100, o 1000 megabits por segundo) y falicidad en su instalación.

Las conexiones más utilizadas se realizan mediante cable de pares trenzados y conectores similares a los telefónicos RJ45. La conexión par trenzado es conocida como 10baseT o 100baseT según la velocidad y utiliza repetidores llamados hubs como puntos de interconexión. La tecnología Ethernet utiliza elementos intermedios de comunicación (hubs, switches, routers) para configurar múltiples segmentos de red y dividir el tráfico para mejorar las prestaciones de transferencia de información.

Existe otro hardware soportado por GNU/Linux para la interconexión de ordenadores, entre los cuales podemos mencionar: Frame Relay o X.25 (utilizada en ordenadores que acceden o interconectan WAN y para servidores con grandes necesidades de transferencias de datos), Packet Radio (interconexión vía radio utilizando protocolos como AX.25, NetRom o Rose) o dispositivos dialing up, que utilizan líneas series, lentas pero muy baratas, a través de módems analógicos o digitales (RDSI, DSL, ADSL, etc.). Estas últimas son las que comúnmente se utilizan en pymes o uso doméstico y requieren otro protocolo para la transmisión de paquetes, tal como SLIP o PPP. Para virtualizar la diversidad de hardware sobre una red, TCP/IP define una interfaz abstracta mediante la cual se concentrarán todos los paquetes que serán enviados por un dispositivo físico (lo cual también significa una red o un segmento de esta red). Por ello, por cada dispositivo de comunicación en la máquina tendremos una interfaz correspondiente en el kernel del sistema operativo.

Las tarjetas Ethernet en GNU/Linux se identifican con **ethx** (donde en todas, x indica un número de orden comenzando por 0), la interfaz a líneas series (módems) se llaman por pppx (para PPP) o slx (para SLIP). Estos nombres son utilizados por los comandos para configurarlas y asignarles el número de identificación que posteriormente permitirá comunicarse con otros dispositivos en la red.

Muchas tarjetas de red están soportadas directamente por el kernel Linux. No obstante, podríamos necesitar cargar el controlador de red para nuestras interfaces como un módulo, esto significa compilar el kernel después de haber escogido el dispositivo adecuado durante el proceso de configuración del kernel.

Los dispositivos de red se pueden mirar en el directorio /dev que es donde existe un archivo (especial, ya sea de bloque o de caracteres según su transferencia), que representa a cada dispositivo hardware.

¿Cómo ver las interfaces de red disponibles?

Ifconfig -a
Este comando muestra todas las interfaces/parámetros por defecto de cada una.

En nuestro equipo encontramos:
- eth0 (primera interfaz)
- eth1(segunda interfaz)
- eth2 (tercera interfaz)
- lo (interfaz de bucle local)
- sit0 (soporte para Ipv6)

2.2.2.3 Componentes de las LAN

Las LANs constan de dos grupos de dispositivos:

Dispositivos de usuario final: Los dispositivos de usuario final incluyen los computadores, impresoras, escáneres, y demás dispositivos que brindan servicios directamente al usuario. También se le conocen con el nombre de **hosts**. Estos dispositivos permiten a los usuarios compartir, crear y obtener información. Los dispositivos hosts pueden existir sin una red, pero sin la red las capacidades de los hosts se ven sumamente limitadas.

Dispositivos de red: son todos aquellos que conectan entre sí a los dispositivos de usuario final posibilitando su intercomunicación. En este grupo tenemos los dispositivos de Networking, las tarjetas de interfaz de red y los medios de transmisión.

Veamos a continuación, dichos dispositivos:

Tarjeta de interfaz de red: Los dispositivos hosts están físicamente conectados con los medios de red mediante una **tarjeta de interfaz de red** (**NIC**). Utilizan esta conexión para realizar las tareas de envío de correo electrónico, impresión de documentos, escaneado de imágenes o acceso a bases de datos. Un NIC es una placa de circuito impreso que se coloca en la ranura de expansión de un bus de la motherboard de un computador, o puede ser un dispositivo periférico. También se denomina adaptador de red. Las NIC para computadores portátiles o de mano por lo general tienen el tamaño de una tarjeta **PCMCIA**. PCMCIA es el acrónimo para Personal Computer Memory Card International Association.

La NIC se comunica con la red a través de una conexión serial y con el computador a través de una conexión paralela. La NIC utiliza una Petición de interrupción (IRQ), una dirección de E/S y espacio de memoria superior para funcionar con el sistema operativo. Un valor IRQ (petición de interrupción) es un número asignado por medio del cual el computador puede esperar que un dispositivo específico lo interrumpa cuando dicho dispositivo envía al computador señales acerca de su operación. Por ejemplo, cuando una impresora ha terminado de imprimir, envía una señal de interrupción al computador. La señal interrumpe momentáneamente al computador de manera que éste pueda decidir qué procesamiento realizar a continuación. Debido a que cada dispositivo necesita una señal distinta para comunicarse con el computador y éste sepa qué dispositivo lo ha interrumpido, se debe especificar un valor único para cada dispositivo y su camino al computador. Antes de la existencia de los dispositvos Plug-and-Play (PnP), los usuarios a menudo tenían que configurar manualmente los valores de la IRQ, o estar al tanto de ellas, cuando se añadía un nuevo dispositivo al computador.

Dispositivos de Red: Los dispositivos de red son los que transportan los datos que deben transferirse entre dispositivos de usuario final. Los dispositivos de red proporcionan el tendido de las conexiones de cable, la concentración de conexiones, la conversión de los formatos de datos y la administración de transferencia de datos. Algunos ejemplos de dispositivos que ejecutan estas funciones son los repetidores, hubs, puentes, switches y routers.

- Un **repetidor** es un dispositivo de red que se utiliza para regenerar una señal. Los repetidores regeneran señales analógicas o digitales que se distorsionan a causa de pérdidas en la transmisión producidas por la atenuación. Un repetidor no toma decisiones inteligentes acerca del envío de paquetes como lo hace un router o puente.

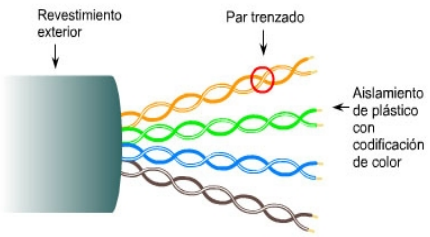
- Los **hubs** concentran las conexiones. En otras palabras, permiten que la red trate un grupo de hosts como si fuera una sola unidad. Esto sucede de manera pasiva, sin interferir en la transmisión de datos. Los hubs activos no sólo concentran hosts, sino que además regeneran señales.

- Los **puentes** convierten los formatos de transmisión de datos de la red además de realizar la administración básica de la transmisión de datos. Los puentes, tal como su nombre lo indica, proporcionan las conexiones entre LAN. Los puentes no sólo conectan las LAN, sino que además verifican los datos para determinar si les corresponde o no cruzar el puente. Esto aumenta la eficiencia de cada parte de la red.

- Los **switches** de grupos de trabajo agregan inteligencia a la administración de transferencia de datos. No sólo son capaces de determinar si los datos deben permanecer o no en una LAN, sino que pueden transferir los datos únicamente a la conexión que necesita esos datos. Otra diferencia entre un puente y un switch es que un switch no convierte formatos de transmisión de datos.

- Los **routers** poseen todas las capacidades indicadas arriba. Los routers pueden regenerar señales, concentrar múltiples conexiones, convertir formatos de transmisión de datos, y manejar transferencias de datos. También pueden conectarse a una WAN, lo que les permite conectar LAN que se encuentran separadas por grandes distancias. Ninguno de los demás dispositivos puede proporcionar este tipo de conexión.

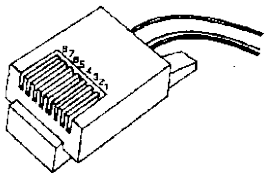
Cableado: El cable de cobre se utiliza en casi todas las LAN. Hay varios tipos de cable de cobre disponibles en el mercado, y cada uno presenta ventajas y desventajas. La correcta selección del cableado es fundamental para que la red funcione de manera eficiente.



El **cable de par trenzado no blindado** (**UTP**) es un medio de cuatro pares de hilos que se utiliza en diversos tipos de redes. Cada uno de los 8 hilos de cobre individuales del cable UTP está revestido de un material aislante. Además, cada par de hilos está trenzado. Este tipo de cable cuenta con un efecto de cancelación de que producen los pares trenzados de hilos para limitar la degradación de la señal que causan la EMI (Electromagnetic Interference) y la RFI (Radiofrecuency Interference). Para reducir aún más la diafonía entre los pares en el cable UTP, la cantidad de trenzados en los pares de hilos varía. El cable Categoría 5e es el que actualmente se recomienda e implementa con mayor frecuencia en las instalaciones.

Para que sea posible la comunicación, la señal transmitida por la fuente debe ser entendida por el receptor. Esto es cierto tanto desde una perspectiva física como en el software. La señal transmitida necesita ser correctamente recibida por la conexión del circuito que está diseñada para recibir las señales. El pin de transmisión de la fuente debe conectarse en fin al pin receptor del destino.

De este modo, para conectar la NIC de un ordenador al switch utilizaremos un cable de *conexión directa.* Por otro lado, para conectar las NIC de dos ordenadores entre sí, tendremos que utilizar un cable UTP llamado *cable de conexión cruzada*. Dicho cable permuta los pines de transmisión y recepción.



Los extremos más utilizados en los cables de par trenzado UTP son los **conectores RJ45**. RJ es un acrónimo inglés de Registered Jack que a su vez es parte del Código Federal de Regulaciones de Estados Unidos. Posee ocho 'pines' o conexiones eléctricas. Es utilizada comúnmente con estándares como EIA/TIA-568B, que define la disposición de los pines o *wiring pinout*.

Medios inalámbricos: La introducción de la tecnología inalámbrica elimina estas limitaciones y otorga portabilidad real al mundo de la computación. En la actualidad, la tecnología inalámbrica no ofrece las transferencias a alta velocidad, la seguridad o la confiabilidad de tiempo de actividad que brindan las redes que usan cables. Sin embargo, la flexibilidad de no tener cables justifica el sacrificio de estas características.

A menudo, los administradores tienen en cuenta las comunicaciones inalámbricas al instalar una nueva red o al actualizar una red existente. Una red inalámbrica puede empezar a funcionar sólo unos pocos minutos después de encender las estaciones de trabajo. Se proporciona la conectividad a Internet a través de una conexión con cable, router, cablemódem o módem DSL y un punto de acceso inalámbrico que sirve de hub para los nodos inalámbricos. En el entorno residencial o de una pequeña oficina, es posible combinar estos dispositivos en una sola unidad.

2.2.2.4 El estándar 802.3

La mayor parte del tráfico en Internet se origina y termina en conexiones de Ethernet. Desde su comienzo en la década de 1970, Ethernet ha evolucionado para satisfacer la creciente demanda de LAN de alta velocidad. En el momento en que aparece un nuevo medio, como la fibra óptica, Ethernet se adapta para sacar ventaja de un ancho de banda superior y de un menor índice de errores que la fibra ofrece. Ahora, el mismo protocolo que transportaba datos a 3 Mbps en 1973 transporta datos a 10 Gbps.

El éxito de Ethernet se debe a los siguientes factores:

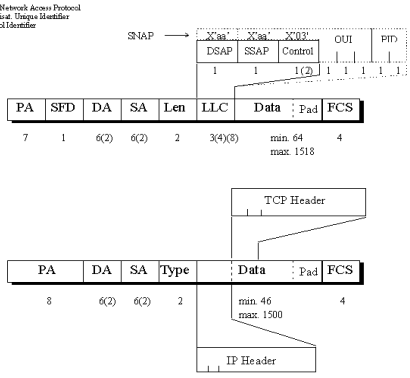
- Sencillez y facilidad de mantenimiento.
- Capacidad para incorporar nuevas tecnologías.
- Confiabilidad
- Bajo costo de instalación y de actualización.

Con la llegada de Gigabit Ethernet, lo que comenzó como una tecnología LAN ahora se extiende a distancias que hacen de Ethernet un estándar de red de área metropolitana (MAN) y red de área amplia (WAN).

La idea original de Ethernet nació del problema de permitir que dos o más hosts utilizaran el mismo medio y evitar que las señales interfirieran entre sí. El problema de acceso por varios usuarios a un medio compartido se estudió a principios de los 70 en la Universidad de Hawai. Se desarrolló un sistema llamado Alohanet para permitir que varias estaciones de las Islas de Hawai tuvieran acceso estructurado a la banda de radiofrecuencia compartida en la atmósfera. Más tarde, este trabajo sentó las bases para el método de acceso a Ethernet conocido como CSMA/CD.

La primera LAN del mundo fue la versión original de Ethernet. Robert Metcalfe y sus compañeros de Xerox la diseñaron hace más de treinta años. El primer estándar de Ethernet fue publicado por un consorcio formado por Digital Equipment Company, Intel y Xerox (DIX). Metcalfe quería que Ethernet fuera un estándar compartido a partir del cual todos se podían beneficiar, de modo que se lanzó como estándar abierto. Los primeros productos que se desarrollaron utilizando el estándar de Ethernet se vendieron a principios de la década de 1980. Ethernet transmitía a una velocidad de hasta 10 Mbps en cable coaxial grueso a una distancia de hasta 2 kilómetros (Km). Este tipo de cable coaxial se conocía como thicknet (red con cable grueso) y tenía el ancho aproximado de un dedo pequeño.

En 1985, el comité de estándares para Redes Metropolitanas y Locales del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) publicó los estándares para las LAN. Estos estándares comienzan con el número 802. El estándar para Ethernet es el **802.3**. El IEEE quería asegurar que sus estándares fueran compatibles con el modelo OSI de la Organización Internacional de Estándares (ISO). Por eso, el estándar IEEE 802.3 debía cubrir las necesidades de la Capa 1 y de las porciones inferiores de la Capa 2 del modelo OSI. Como resultado, ciertas pequeñas modificaciones al estándar original de Ethernet se efectuaron en el 802.3. Las diferencias entre los dos estándares fueron tan insignificantes que cualquier tarjeta de interfaz de la red de Ethernet (NIC) puede transmitir y recibir tanto tramas de Ethernet como de 802.3. Básicamente, Ethernet y IEEE 802.3 son un mismo estándar.



El ancho de banda de 10 Mbps de Ethernet era más que suficiente para los lentos computadores personales (PC) de los años 80. A principios de los 90, los PC se volvieron mucho más rápidos, los tamaños de los archivos aumentaron y se producían cuellos de botella en el flujo de los datos. La mayoría a causa de una baja disponibilidad del ancho de banda. En 1995, el IEEE anunció un estándar para la Ethernet de 100 Mbps. Más tarde siguieron los estándares para Ethernet de un gigabit por segundo (Gbps, mil millones de bits por segundo) en 1998 y 1999.

Todos los estándares son básicamente compatibles con el estándar original de Ethernet. Una trama de Ethernet puede partir desde una antigua NIC de 10 Mbps de cable coaxial de un PC, subir a un enlace de fibra de Ethernet de 10 Gbps y terminar en una NIC de 100 Mbps. Siempre que permanezca en redes de Ethernet, el paquete no cambia. Por este motivo, se considera que Ethernet es muy escalable. El ancho de banda de la red podría aumentarse muchas veces sin cambiar la tecnología base de Ethernet.

2.2.3 Redes inalámbricas WLAN

2.2.3.1 Introducción a las WLAN

Una red de área local o WLAN (Wireless LAN) utiliza ondas electromagnéticas (radio e infrarrojo) para enlazar (mediante un adaptador) los equipos conectados a la red, en lugar de de los cables de coble o de fibra óptica que se utilizan en las LAN convencionales cableadas (Ethernet, Token Ring, etc...)

Las redes locales inalámbricas más que una sustitución de las LANs convencionales son una extensión de las mismas, ya que permiten el intercambio de información entre los distintos medios en una forma transparente al usuario.

En este sentido el objetivo fundamental de las redes WLAN es el de proporcionar las facilidades no disponibles en los sistemas cableados y formar una red donde coexistan los dos tipos de sistemas. Enlazando los diferentes equipos o terminales móviles asociados a la red.

Este hecho proporciona al usuario una gran movilidad sin perder conectividad. El atractivo fundamental de este tipo de redes es la facilidad de la instalación y el ahorro que supone la supresión del medio de transmisión cableado. Las redes inalámbricas son la alternativa ideal para hacer llegar una red tradicional a lugares donde el cableado no lo permite. En general las WLAN se utilizarán como complemento de las redes fijas.

2.2.3.2 El estándar 802.11

El estándar **IEEE 802.11** fue desarrollado por el IEEE (Institute of Electrical and Electronics Engineers, Instituto de Ingenieros Eléctricos y Electrónicos) organismo internacional dedicado a la estandarización en telecomunicaciones. El objetivo del estándar IEEE 802.11, también conocido por **Wi-Fi** (abreviatura de Wireless Fidelity) es la interconexión inalámbrica de ordenadores a nivel local, pero por su evolución, tambien permite otros servicios, como el acceso a Internet y la interconexión a mayor distancia.

Wi-Fi es una marca de la Wi-Fi Alliance, organismo que comprueba y certifica el cumplimiento de los estándares IEEE 802.11 en los equipos comerciales. Wi-Fi engloba todos los estándares **IEEE 802.11**, al ser varios, tambien se conoce como **IEEE 802.11x.**

IEEE 802.11x es el estándar de protocolo de comunicaciones que define el uso de los dos niveles más bajos de la arquitectura OSI (capa de enlace de datos y capa física). Este estándar define las normas de funcionamiento para redes de area local inalámbricas o WLAN (Wireless Local Area Network).

El estándar IEEE 802.11 se creo en 1997 (estándar original) para redes inalámbricas que empleaban microondas (frecuencia de trabajo de 2,4 GHz) y velocidades de transmisión entre 1 y 2 Mbps.

En la siguiente tabla resumimos los estándares de 802.11 existentes:

Estándar IEEE	***Finalización***	***Qué define***
802.11	1997	Estandar inicial de las WLAN, infrarrojos, banda 2,4Ghz. 1 y 2Mbps. Acceso CMSA/CD
802.11b	1999	11Mbps Banda 2,4Ghz. Acceso CMSA/CD
802.11a	1999	WLAN de alta velocidad, banda 5 Ghz, 54Mbps. Necesita puntos de vista.
802.11h	2003	Técnicas de gestión de espectro para 802.11a.
802.11g	2003	WLAN de alta velocidad alternativo, banda 2,4Ghz. 54Mbps. SuperG: 108Mbps (propietario)
802.11i	2004	Funciones de seguridad específicas para WLAN.
802.11n	2006	Utiliza tecnología MIMO, para alcanzar velocidades “teóricas” de 500MBps

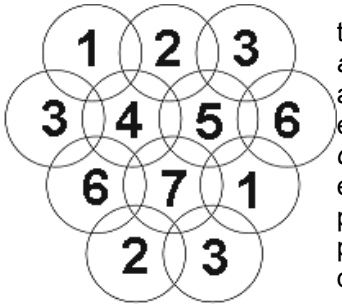
Topologías

La versatilidad y flexibilidad de las redes inalámbricas es el motivo por el cual la complejidad de una LAN implementada con esta tecnología sea tremendamente variable. Esta gran variedad de configuraciones ayuda a que este tipo de redes se adapte a casi cualquier necesidad.

Estas configuraciones se pueden dividir en dos grandes grupos, las redes peer to peer y las que utilizan puntos de acceso.

- Peer to peer: también conocidas como **redes ad-hoc**, es la configuración más sencilla, ya que en ella los únicos elementos necesarios son terminales móviles equipados con los correspondientes adaptadores para comunicaciones inalámbricas. En este tipo de redes, el único requisito deriva del rango de cobertura de la señal, ya que es necesario que los terminales móviles estén dentro de este rango para que la comunicación sea posible. Por otro lado, estas configuraciones son muy sencillas de implementar y no es necesario ningún tipo de gestión administrativa de la red.

- **Punto de acceso:** Estas configuraciones utilizan el **concepto de celda,** ya utilizado en otras comunicaciones inalámbricas, como telefonía móvil. Una celda podría entenderse como el área en la que una señal radioeléctrica es efectiva. A pesar de que en el caso de las redes inalámbricas esta celda suele tener un tamaño reducido, mediante el uso de varias fuentes de emisión es posible combinar las celdas de estas señales para cubrir de forma casi total un área más extensa.



La estrategia empleada para aumentar el número de celdas, y por lo tanto el área cubierta por la red, es la utilización de los llamados *Puntos de acceso*, que funcionan como repetidores, y por tanto son capaces de doblar el alcance de una red inalámbrica, ya que ahora la distancia máxima permitida no es entre estaciones, sino entre una estación y un punto de acceso. Los *Puntos de acceso* son colocados normalmente en alto, pero solo es necesario que estén situados estratégicamente para que dispongan de la cobertura necesaria para dar servicio a los terminales que soportan. Un único punto de acceso puede soportar un pequeño grupo de usuarios y puede funcionar en un rango de al menos treinta metros y hasta varios cientos de metros.

La técnica de *Punto de acceso* es capaz de dotar a una red inalámbrica de muchas más posibilidades. Además del evidente aumento del alcance de la red, ya que la utilización de varios puntos de acceso, y por lo tanto del empleo de varias celdas que colapsen el lugar donde se encuentre la red, permite lo que se conoce como *roaming*, es decir que los terminales puedan moverse sin perder la cobertura y sin sufrir cortes en la comunicación. Esto representa una de las características más interesantes de las redes inalámbricas.

Para nuestras pruebas trabajaremos con un punto de acceso DWL-G700AP de D-Link.

Dicho punto de acceso soporta encriptación WEP, WPA y 802.1x.
Y velocidades de transmisión de: 1,2,5'5,11,6,9,12,18,24,36,48,54Mbps.

Capa Física

Las tecnologías inalámbricas radiofrecuencia utilizadas en las WLAN son DSSS (Direct Sequence Spread Spectrum) y FHSS (Frecuency Hopping Spread Spectrum).

FHSS: La tecnología de espectro ensanchado por salto en frecuencia (FHSS) consiste en transmitir una parte de la información en una determinada frecuencia durante un intervalo de tiempo llamada *dwell time* e inferior a 400 ms. Pasado este tiempo se cambia la frecuencia de emisión y se sigue transmitiendo a otra frecuencia. De esta manera cada tramo de información se va transmitiendo en una frecuencia distinta durante un intervalo muy corto de tiempo.

El orden en los saltos en frecuencia se determina según una secuencia pseudoaleatoria almacenada en unas tablas, y que tanto el emisor y el receptor deben conocer. Si se mantiene la sincronización en los saltos de frecuencias se consigue que, aunque en el tiempo se cambie de canal físico, a nivel lógico se mantiene un solo canal por el que se realiza la comunicación.

Esta técnica también utiliza la zona de los 2.4GHz, la cual organiza en 79 canales con un ancho de banda de 1MHz cada uno. El número de saltos por segundo es regulado por cada país, así, por ejemplo, Estados Unidos fija una tasa mínima de saltos de 2.5 por segundo.

El estándar IEEE 802.11 define la modulación aplicable en este caso. Se utiliza la modulación en frecuencia FSK (Frequency Shift Keying), con una velocidad de 1Mbps ampliable a 2Mbps. En la revisión del estándar, la 802.11b, esta velocidad también ha aumentado a 11Mbps. La técnica FHSS seria equivalente a una multiplexación en frecuencia.

DSSS: En esta técnica se genera un patrón de bits redundante (señal de chip) para cada uno de los bits que componen la señal. Cuanto mayor sea esta señal, mayor será la resistencia de la señal a las interferencias. El estándar IEEE 802.11 recomienda un tamaño de 11 bits, pero el optimo es de 100. En recepción es necesario realizar el proceso inverso para obtener la información original.

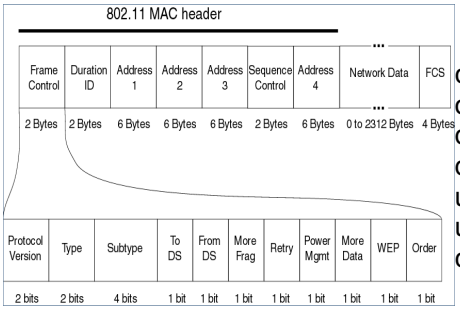
La secuencia de bits utilizada para modular los bits se conoce como secuencia de Barker (también llamado código de dispersión o PseudoNoise). Es una secuencia rápida diseñada para que aparezca aproximadamente la misma cantidad de 1 que de 0. Un ejemplo de esta secuencia es el siguiente. +1-1+1+1-1+1+1+1-1-1-1-1 Solo los receptores a los que el emisor haya enviado previamente la secuencia podrán recomponer la señal original. Además, al sustituir cada bit de datos a transmitir, por una secuencia de 11 bits equivalente, aunque parte de la señal de transmisión se vea afectada por interferencias, el receptor aún puede reconstruir fácilmente la información a partir de la señal recibida.

En el caso de Estados Unidos y Europa la tecnología DSSS utiliza un rango de frecuencias que va desde los 2,4 GHz hasta los 2,4835 GHz, lo que permite tener un ancho de banda total de 83,5 MHz. Este ancho de banda se subdivide en canales de 5 MHz, lo que hace un total de 14 canales independientes. Cada país está autorizado a utilizar un subconjunto de estos canales. En el caso de España se utilizan los canales entre 1 y 11, preferentemente los canales 1, 6 y 11 para evitar interferencias.

En configuraciones donde existan más de una celda, éstas pueden operar simultáneamente y sin interferencias siempre y cuando la diferencia entre las frecuencias centrales de las distintas celdas sea de al menos 30 MHz, lo que reduce a tres el número de canales independientes y funcionando simultáneamente en el ancho de banda total de 83,5 MHz. Esta independencia entre canales nos permite aumentar la capacidad del sistema de forma lineal

Capa de Acceso al medio

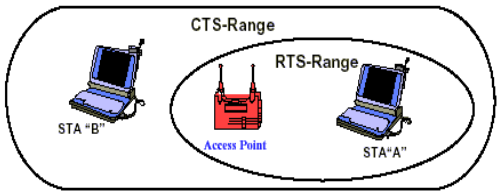
La tecnología de acceso al medio está formada por dos subcapas: “Logical Link Control (LLC)” y “Media Access Control” (MAC). 802.3 MAC es muy similar a 802.3 en el sentido que tenemos varios usuarios compartiendo un medio.



En 802.3 Ethernet, el método (CSMA/CD) se encarga de controlar quién accede al medio y evitar y detectar colisiones con otros equipos que transmiten simultáneamente. En 802.11 la detección de colisión no es posible porque para ello una estación debe transmitir y recibir al mismo tiempo. Por tanto, en 802.11 se utiliza (CSMA/CA), cuando se recibe un paquete correcto se devuelve un ACK para confirmar la entrega del paquete y que no ha habido colisión.

Como hemos visto CSMA/CA proporciona un método de compartir el aire, este mecanismo de ACK tiene más sobrecarga de transmisión que 802.3; en consecuencia, las redes 802.11 serán siempre más lentas que las Ethernet LAN.

Otro problema específico de la capa MAC es el llamado “nodo oculto”, en el que dos estaciones en lados opuestos a un punto de acceso pueden oir las señales que él está transmitiendo, pero no la estaciones entre sí. Para solucionar este problema la norma 802.11 especifica unas señales opcionales llamadas RTS/CTS. Cuando se utiliza, una estación transmite un RTS al punto de acceso y espera a que éste le mande un CTS. Como la señal CTS la oyen todos las estaciones, éstas esperarán para transmitir su petición y no causar colisión con la estación que mandó el RTS.



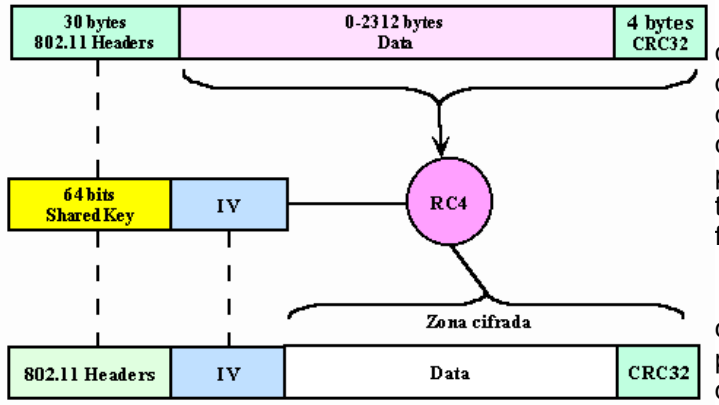
Por último, la capa MAC 802.11 proporciona CRC Checksum y fragmentación de paquetes.

2.2.3.3 Encriptación WEP y WPA

WEP (Wired Equivalent Privacy) fue el primer protocolo de encriptación introducido en el primer estándar 802.11 allá por 1999. Está basado en el algoritmo de encriptación RC4, con una clave secreta de 40 o 104 bits, combinada con un vector de inicialización (IV) de 24 bits para encriptar el mensaje de texto M y su checksum, el ICV (Integrity Check Value). El mensaje encriptado C se determinaba utilizando la siguiente fórmula:

$$C = [\, M \,||\, ICV(M) \,] + [\, RC4\,(K \,||\, IV) \,]$$

Donde || es un operador de concatenación y + es un operador XOR.



Claramente, el vector de inicialización es la clave de la seguridad WEP, es generado dinámicamente y debería ser diferente para cada trama. El objetivo perseguido con el IV es cifrar con claves diferentes para impedir que un posible atacante pueda capturar suficiente tráfico cifrado con la misma clave y terminar finalmente deduciendo dicha clave.

Como es lógico, ambos extremos deben conocer tanto la clave secreta como el IV. Lo primero lo sabemos ya que es conocido puesto que está almacenado en la configuración de cada elemento de red.

El IV, en cambio, se genera en un extremo y se envía en la propia trama al otro extremo, por lo que también será conocido. Observemos que al viajar el IV en cada trama es sencillo de interceptar por un posible atacante.

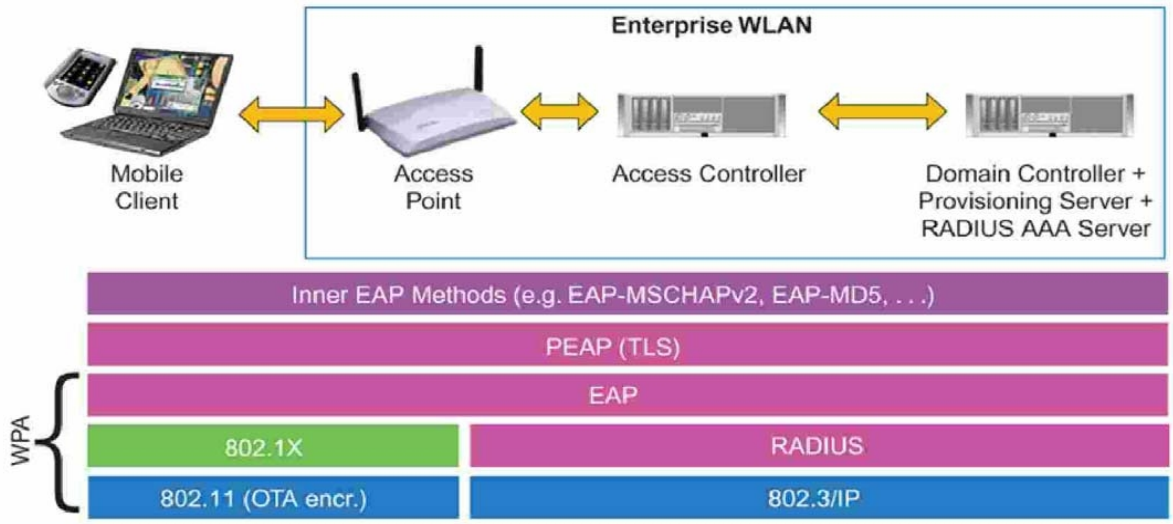
Los fallos de seguridad de WEP pueden resumirse tal y como sigue:

- Debilidades del algoritmo RC4 dentro del protocolo WEP debido a la construcción de la clave.
- Los IVs son demasiado cortos (24 bits – hacen falta menos de 5000 paquetes para tener un 50% de posibilidades de dar con la clave) y se permite la reutilización de IV (no hay protección contra la repetición de mensajes).
- No existe una comprobación de integridad apropiada (se utiliza CRC32 para la detección de errores y no es criptográficamente seguro por su linealidad).
- No existe un método integrado de actualización de las claves.

WPA (WiFi Protected Access) fue diseñado para utilizar un servidor de autenticación como RADIUS, que distribuye claves diferentes a cada usuario (a través del protocolo 802.1x); sin embargo, también se puede utilizar en un modo menos seguro de clave pre-compartida ([PSK]-PreSharedKey) para usuarios de casa o pequeña oficina. La información es cifrada utilizando el algoritmo RC4 (debido a que WPA no elimina el proceso de cifrado WEP, sólo lo fortalece), con una clave de 128 bits y un vector de inicialización de 48 bits.

Una de las mejoras sobre WEP, es la implementación del **Protocolo de Integridad de Clave Temporal** (TKIP – *Temporal Key Integrity Protocolo*), que cambia claves dinámicamente a medida que el sistema es utilizado. Cuando esto se combina con un vector de inicialización (IV) mucho más grande, evita los ataques de recuperación de clave (ataques estadísticos) a los que es susceptible WEP.

Adicionalmente a la autenticación y cifrado, WPA también mejora la integridad de la información cifrada. El chequeo de reduncancia cíclica (CRC – Cyclic Redundancy Check) utilizado en WEP es inseguro, ya que es posible alterar la información y actualizar el CRC del mensaje sin conocer la clave WEP. WPA implementa un código de integridad del mensaje (MIC – Message Integrity Code), también conocido como "Michael". Además, WPA incluye protección contra ataques de "repetición" (replay attacks), ya que incluye un contador de tramas.



Al incrementar el tamaño de claves, el número de llaves en uso, y al agregar un sistema de verificación de mensajes, WPA hace que la entrada no autorizada a redes inalámbricas sea mucho más difícil. El algoritmo Michael fue el más fuerte que los diseñadores de WPA pudieron crear, bajo la premisa de que debía funcionar en las tarjetas de red inalámbricas más viejas; sin embargo es suceptible a ataques. Para limitar este riesgo, las redes WPA se desconectan durante 60 segundos al detectar dos intentos de ataque durante 1 minuto.

WPA2 está basada en el nuevo estándar 802.11i. WPA, por ser una versión previa, que se podría considerar de "migración", no incluye todas las características del IEEE 802.11i, mientras que WPA2 se puede inferir que es la versión certificada del estándar 802.11i. El estándar fue ratificado en Junio de 2004. La alianza Wi-Fi llama a la versión de clave pre-compartida WPA-Personal y WPA2-Personal y a la versión con autenticación 802.1x/EAP como WPA-Enterprise y WPA2-Enterprise.

2.2.3.4 Configuración de un cliente Wireless en Linux

En GNU/Linux existen herramientas para utilizar tarjetas inalámbricas. Las herramientas fundamentales utilizan una configuración en modo texto quizás algo tediosa para usuarios noveles; sin embargo, también hay otras herramientas para entornos visuales como Gnome o KDE que presentan interfaces amigables para gestionar los parámetros de nuestra conexión Wireless.

En este apartado nos centraremos exclusivamente en cómo configurar una estación cliente que corre en GNU/Linux (ya bien sea una estación de escritorio o un servidor sin intefaz gráfico) para conectarse a un punto de acceso inalámbrico. Por consiguiente, todos los comandos que vamos a ver no los vamos a ir probando en nuestro dispositivo Debian, sino en un ordenador distinto (que en nuestro caso particular corre sobre una distribución Ubuntu). Ya veremos más adelante, cuando lleguemos a la parte de control de acceso, qué papel jugará nuestro dispositivo Debian en este escenario de red sin cables.

Para poder usar tarjetas de red inalámbricas necesitamos instalar el paquete wireless-tools:

```
apt-get install wireless-tools
```

No hay que confundir las Wireless Tools con las Wireless-Extensions. Las Wireless tools son una serie de utilidades que nos permiten configurar nuestra tarjeta inalámbrica mientras que las Wireless extensions es una API para desarrolladores de drivers que se encuentra integrada en el kernel.

En la siguiente tabla resumimos las herramientas de Wireless Tools:

Comando	***Finalización***
iwconfig	Es similar a *ifconfig*. Se utilizada para configurar un interfaz de red inalámbrico.
iwlist	Obtiene información más detallada sobre el interfaz inalámbrico.
iwspy	Obtiene estadísticas sobre calidad del enlace de nodos inalámbricos específicos en una red.
iwpriv	Configura parámetros opcionales del interfaz.
iwevent	Muestra eventos generados por el dispositivo inalámbrico.
iwgetid	Muestra información sobre la conexión actual (ESSID, AP address..)

A continuación haremos uso de los comandos anteriores para conectar nuestra tarjeta *Wireless* a la red inalámbrica. El resultado de ejecutar *iwconfig* en nuestra consola es:

```
lo        no wireless extensions.

eth0      no wireless extensions.

sit0      no wireless extensions.

eth1      IEEE 802.11g  ESSID:off/any
          Mode:Managed  Frequency:2.462 GHz  Access Point: Not-Associated
          Bit Rate:2 Mb/s   Tx-Power:32 dBm
          RTS thr:2347 B   Fragment thr:2346 B
          Power Management:off
          Link Quality:0  Signal level:0  Noise level:0
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Para ver qué redes inalámbricas hay a nuestro alcance utilizaremos el comando *iwlist*. El resultado en nuestro equipo es:

```
          Cell 01 – Address: 00:15:E9:F3:CB:25
          ESSID: “radius”
          Protocol:IEEE 802.11g
          Mode:Managed
          Frequency:2.442 GHz (Channel 6)
          Quality: 0/100 Signal level:-53 dBm  Noise level:-256 dBm
          Encryption key:on
          Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6 Mb/s
                    9 Mb/s; 12 Mb/s; 18 Mb/s; 36 Mn/s;
                    48 Mb/s; 54 Mb/s
          Extra: bcn_int=119
          Extra:atim=3
```

Observamos que radius es el ESSID de nuestro punto de acceso, que está transmitiendo por el canal 6 y lo hace de forma encriptada.

Configurando el driver Ndiswrapper

Todos sabemos que uno de los mayores problemas de Linux ha sido siempre la falta de drivers específicos para todo el hardware disponible. La falta de compromiso de los fabricantes de nuevos drivers (justificada en parte por el escaso porcentaje de usuarios en Linux) ha dejado en sombras a muchos usuarios de Linux, hasta el punto de que es normal preguntarse al elegir un nuevo componente para nuestro ordenador si podemos "echarlo a andar" en Linux.

La verdad es que siempre, con más o menos esfuerzo, se consiguen solucionar la mayoría de los problemas. El avanzado desarrollo del kernel en este sentido y la adaptación por parte de las distintas distribuciones Linux de los drivers que se desarrollan paralela y altruístamente para estos componentes facilita las cosas.

Distinto es el caso de las tarjetas de red Wireless. La tecnología Wi-Fi lleva tan poco tiempo en el mercado que es muy normal que no encontremos drivers para todos los modelos. La mayoría de los fabricantes únicamente ofrece drivers para las últimas versiones de Windows (WinXP ó Windos Vista). En este sistema operativo, la mayoría de los drivers de las tarjetas usan llamadas al protocolo NDIS de Microsoft. Es decir, que crean drivers que se comunican con el sistema operativo de una forma genérica dictada por este protocolo. Aquí entra en juego la utilidad de NdisWrapper.

Como su propio nombre indica, NdisWrapper lo que hace es mapear los drivers NDIS de WinXP/Win2000 de cada fabricante a un módulo genérico en el kernel de Linux. Es decir, que NdisWrapper usa los drivers propietarios en Windows de cada tarjeta y transforma sus llamadas al protocolo NDIS a llamadas a un módulo Wireless cargado en el Kernel de Linux. Con esta utilidad conseguiremos hacer funcionar sin ningún problema casi cualquier tarjeta wireless, pero además más dispositivos que usan el NDIS. El proyecto tiene un sistema de documentación basado en wiki estupendo.

Para instalar NdisWrapper:

```
apt-get install ndiswrapper-utils
```

A continuación instalaremos el driver de nuestra tarjeta Wireless:

El modelo exacto que tenemos en el laboratorio es "Sitecom Wireless Network PCI Card 54Mbps WL121v2"

Para ver qué driver utiliza Linux ejecutamos: lshw -C network
El resultado es el siguiente:

```
  *-network:1
       description: Wireless interface
       product: ISL3886 [Prism Javelin/Prism Xbow]
       vendor: Intersil Corporation
       physical id: 4
       bus info: pci@03:04.0
       logical name: eth1
       version: 01
       serial: 00:0c:f6:07:82:b9
       width: 32 bits
       clock: 33MHz
       capabilities: bus_master cap_list ethernet physical wireless
       configuration: broadcast=yes driver=islsm_pci link=no multicast=yes wireless=IEEE 802.11g
       resources: iomemory:cfffc000-cfffdfff irq:169
```

Observamos que el driver que Linux utiliza es islsm_pci, pero con este driver no podemos controlar nuestra tarjeta, ya que vemos como despúes de haber intentado varias veces conectarnos a nuestro punto de acceso haciendo uso de las Wireless-tools la tarjeta no hace nada, no se llega a asociar al punto de acceso. Por tanto llegamos a la conclusión que deberíamos instalar ndriswrapper con su driver correspondiente en Windows.

Lo primero es eliminar el actual driver islsm_pci mediante:

```
modprobe -r islsm_pci
```

Modprobe es un programa que sirve para añadir y eliminar modulos del kernel de Linux. Para cargar un módulo se utiliza modprobe seguido del módulo a cargar y para eliminar se utiliza la opción -r. Como hemos comentado anteriormente, el kernel de Linux tiene un diseño modular. Cuando un usuario solicita alguna característica que no está presente en el kernel residente, se carga dinámicamente en memoria un módulo kernel, también conocido algunas veces como controlador. Si se añade un nuevo hardware después de la instalación y éste requiere un módulo kernel, el sistema debe ser configurado para cargar el módulo adecuado para el nuevo hardware. Por defecto, modprobe intenta cargar el módulo desde los subdirectorios /lib/modules/<kernel-version>/kernel/drivers/. Hay un subdirectorio para cada tipo de módulo, tal como el subdirectorio net/ para los controladores de interfaces de red.

Ahora descargamos de Internet los drivers para WinXP de nuestra tarjeta, (en nuestro caso lo descargamos de: http://support.fujitsu-siemens.com/download/ShowDescription.asp?SoftwareGUID=3B928095-EEEE-4194-91A7-E0CBE2D28B56&ClassID=0090118B-10DE-46FC-BA66-C1BF106799C5)

Al extraer el paquete encontramos dos archivos: PRISMA00.inf y PRISMA00.sys
Con Ndiswrapper instalaremos nuestro driver de WinXP utilizando la sintaxis:

ndiswrapper -i nomber_del_driver.inf

```
ndiswrapper -i PRISMA00.inf

ndiswrapper -l
        Installed ndis drivers:
        prisma00            driver present, hardware present
```

Al ejecutar ndiswrapper -l nos salen las tarjetas cuyos drivers Win hayamos instalado en NdisWrapper (pueden ser más de una). Si la tarjeta está físicamente instalada, debería decir algo como lo que nos aparece en el cuadro anterior.

El siguiente paso es insertar el módulo genérico NdisWrapper en el kernel:

```
modprobe ndiswrapper
```

Para cargar ndiswrapper al arrancar:

```
ndiswrapper -m
```

Si no ha fallado ya tenemos nuestra tarjeta reconocida e instalada.
Utilizaremos el comando *ifconfig* para establecer parámetros de red (capa IP) e iwconfig para párametros del nivel de enlace (wireless)

Conexión al punto de acceso utilizando cifrado WEP

Una vez configurado el driver de nuestra tarjeta Wireless procedemos a conectarnos a nuestro punto de acceso en el que estamos utilizando una clave WEP.

Utilizaremos en este caso el comando *iwconfig*. La sintaxis general de este comando es:
iwconfig interfaz acción parámetro

Veamos algunas de las opciones que ofrece:

- **iwconfig eth1 essid NombreDeRed:** De esta manera asociamos nuestra interfaz eth1 con una red virtual formada por un conjunto de puntos de acceso identificados por el essid o nombre de red. Si el nombre de red proporcionado es la cadena vacía, esto es, la cadena “”, la interfaz se asociará con el mejor punto de acceso disponible, con el cual tengamos una señal más fuerte. Después de ejecutar la orden tendremos acceso a la LAN que hay detrás del punto de acceso con el que se conecte.

- **iwconfig eth1 mode {adhoc, managed, master, monitor}:** Un adaptador inalámbrico puede trabajar en varios modos que determinan la forma en que se comunica con los demás equipos de la red. Un punto de acceso trabaja en modo master y nosotros utilizaremos el modo managed para contactar con ese punto de acceso. El modo ad-hoc se utiliza cuando queremos establecer comunicación con otro equipo directamente sin necesidad de un punto de acceso intermedio.

- **iwconfig eth1 key s:clave:** Para proteger nuestras comunicaciones las cifraremos mediante el protocolo WEP definido en el estándar 802.11 asignando una clave con este comando. Para la clave usaremos cinco caracteres ASCII para cifrado de 40 bits y 13 caracteres para cifrado de 104 bits. O bien introduciendo la clave directamente en hexadecimal sin s: delante.

- **iwconfig eth1:** Muestra información sobre el enlace. Podemos ver el nombre de red (ESSID), el modo en el que está nuestra tarjeta, el punto de acceso que estamos utilizando y algunas cosas más. Uno de los valores interesantes es la calidad del enlace que nos da una idea de la calidad del servicio que puede ofrecernos el punto de acceso.

  Por defecto nuestra tarjeta detectará el punto de acceso más "cercano" y se asociará con él. De todas maneras podemos ver qué redes hay disponibles ahí fuera con iwlist.

  Una vez establecidos los parámetros WiFi con *iwconfig*, procederemos a configurar la tarjeta de red en la forma habitual, con los comandos *ifconfig* y *route*.

  Si tecleamos *iwconfig eth1* obtenemos por pantalla:

```
eth1    IEEE 802.11g   ESSID:"radius"
        Mode:Managed   Frequency:2.437 GHz Access Point: 00:15:E9:F3:CB:25
        Bit Rate:18 Mb/s    Tx-Power:32 dBm
        RTS thr:2347 B    Fragment thr:2346 B
        Power Management:off
        Link Quality:100/100  Signal level:-73 dBm  Noise level:-256 dBm
        Rx invalid nwid:0   Rx  invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0   Invalid misc:0    Missed beacon:0
```

Conexión al punto de acceso utilizando cifrado WPA

Para llevar a cabo esta tarea haremos uso de **wpa_supplicant**. Wpa_supplicant es una implementación software libre de IEEE 802.11i. Es un paquete que nos permite conectarnos a puntos de acceso con cifrado WPA. Soporta WPA, WPA-PSK, WPA2-PSK (WPA Personal), cifrado con claves CCMP, TKIP, RSN...

Para instalarlo en nuestra máquina Linux haremos simplemente:

```
apt-get install wpasupplicant
```

Necesitaremos configurar wpa_supplicant adecuadamente para hacerlo funcionar con nuestro punto de acceso. El fichero de configuración de wpasupplicant consiste en una lista de paŕametros de red globales para cada red inalámbrica con distintos SSID. Dichos parámetros se enumeran en el fichero de configuración con una sintaxis como la siguiente:

parametro=valor

Nuestro fichero de configuración viene a ser tal que así:

```
ctrl_interface=/var/run/wpa_supplicant

network={
      ssid="radius"
      proto=WPA RSN
      key_mgmt=WPA-PSK
      psk="password"
}
```

Donde *ctrl_interface* es el nombre del path donde wpa_supplicant crea ficheros de comunicación para integrarse con programas front_ends.

Las redes se configuran en "bloques" que comienzan por
network={
    parameter=value
    ...
}

Donde:

- **ssid:** es el nombre de la red.
- **proto:** es el protocolo utilizado.
- **key_mgmt**: es el protocolo utilizado para el juego de claves. Nuestro protocolo es WPA-PSK (PreShared Keys).
- **psk:** es la clave utilizada.

Para conocer todos los parámetros de configuración existentes utilizaremos:

```
man wpa_supplicant.conf
```

Para arrancar wpa_supplicant cada vez que activemos la interfaz tendremos que editar el archivo */etc/network/interfaces* y escribir una configuración similar a:

```
auto eth1
iface eth1 inet dhcp
wireless-essid radius
pre-up wpa_supplicant -B -ieth1 -c/etc/wpa_supplicant.conf
post-down killall -q wpa_supplicant
```

La primera línea nos dice que la interfaz eth1 se activará automáticamente. La línea dos que tendrá una dirección IP dinámica (Hablaremos de DHCP en el siguiente capítulo). La tercera línea que la interfaz se asociará a la red inalámbrica con identificador radius. La cuarta activa wpa_supplicant y la última ejecuta el comando killall (destruir proceso) cuando se desactiva la interfaz.

Como vemos wpa_supplicant está seguido de una serie de parámetros:
-B : el demonio corre en background.
-i : interfaz donde escucha.
-c : lugar donde reside el archivo de configuración.

Finalmente tras ejecutar *ifconfig eth1 up* estaremos felizmente conectados utilizando nuestra conexión wpa.

2.2.3.5 Deficiencias en la seguridad

Como hemos comentado, las redes inalámbricas se comunican mediante un medio compartido, es decir, todos los dispositivos asociados a la red son capaces de "ver" todos los datos que se transmiten por la red, les vayan destinados o no, e incluso pueden suplantar la personalidad de otro dispositivo. Para tratar de paliar estos problemas, típicamente se han implantado ciertos mecanismos para asegurar la autenticación ante la red y el cifrado de los datos, como ya hemos visto, a la hora de transmitir. Estos mecanismos son:

- **Listas de control de acceso basadas en direcciones MAC**

  Una de las medidas más comunes que se utilizan para securizar una red Wireless es restringir las máquinas que podrán comunicarse con el punto de acceso haciendo filtrado por dirección MAC en éste. Para esto se suele crear una tabla en el punto de acceso que contiene todas las MACs de los clientes que están autorizados para conectar. Aunque esto pueda parecer una medida de seguridad efectiva, no lo es, ya que es muy fácil cambiar la dirección MAC que aparece en los paquetes que un cliente envía, y hacernos pasar por uno de los equipos que sí tienen acceso a la red.

- **No emitir Becaon Frames (o emitirlos sin el ESSID)**

  Una medida de seguridad bastante común es “ocultar” el ESSID (el nombre de la red), es decir, hacer que el AP (Access Point) no mande beacon frames, o en su defecto no incluya el ESSID en éstos. En este caso, para descubrir el ESSID solo habría que capturar datos de la red y esperar a que un cliente se conectara, y veríamos el ESSID en la trama Probe Request del cliente (en el caso de que no se manden Beacon Frames), o en la trama Probe Response del punto de acceso.

- **Utilizar 802.1x para la autenticación ante la red**

  Este sistema de autenticación obliga a instalar un programa en el ordenador que haga de cliente de 802.1x si el sistema operativo no lo soporta de forma nativa, y no soluciona el problema del cifrado de los datos, ya que sólo es una solución para la autenticación.

- **Utilizar WEP para cifrar los datos**

  Para proteger los datos que se envían a través de las WLANs, el estándar 802.11b define el uso del protocolo WEP (Wired Equivalent Privacy). WEP intenta proveer la seguridad de una red con cables a una red Wireless, encriptando los datos que viajan sobre las ondas radioeléctricas en las dos capas más bajas del modelo OSI (capa física y capa de enlace). El protocolo WEP está basado en el algoritmo de encriptación RC4, y utiliza claves de 64bits o de 128 bits, que en realidad son de 40 y 104bits, ya que los otros 24 bits van en el paquete como Vector de Inicialización (IV).

  Aunque esto pueda parecer suficientemente seguro, no lo es, pues se han encontrado numerosas vulnerabilidades en el mecanismo de encriptación que hacen desaconsejable su uso, ya que sólo hay que capturar el tráfico (que viaja por un medio compartido) y desencriptarlo con alguna de las herramientas ampliamente difundidas por Internet para tal fin.

- **UNA SOLUCIÓN: Redes Privadas Virtuales**

  Se basa en una red 802.11 abierta en la que se obtiene una dirección IP privada por DHCP al conectar y posteriormente se establece un túnel seguro (previa autenticación) con un servidor dedicado. Este túnel se establece mediante el protocolo estándar IPSec, cuya fiabilidad y seguridad han sido sobradamente probadas, con lo que se resuelven los problemas de autenticación y cifrado de datos.

- **OTRA SOLUCIÓN: WPA + 802.1x**

  WPA (Wi-Fi Protected Access) es un sistema para proteger las redes inalámbricas creado para corregir las deficiencias del WEP comentadas anteriormente. Está diseñado para utilizar un servidor de autenticación (normalmente un servidor RADIUS), que distribuye claves diferentes a cada usuario mediante el protocolo 802.1x. La información se cifra utilizando el algoritmo RC4, una clave de 128 bits y un vector de inicialización de 48 bits (WPA no elimina el proceso de cifrado WEP, sólo lo fortalece). Una de las mejoras sobre WEP, es la implementación del Protocolo de Integridad de Clave Temporal (TKIP, Temporal Key Integrity Protocol), que cambia las claves dinámicamente a medida que el sistema es utilizado. Cuando esto se combina con un vector de inicialización (IV) mucho más grande, evita los ataques de recuperación de clave a los que es susceptible WEP.

  Esta es la solución que implementaremos en nuestro dispositivo y que, actualmente, es la forma de acceso a red inalámbrica recomendada, dado que es una solución más ligera que las Redes Privadas Virtuales e infranqueable.

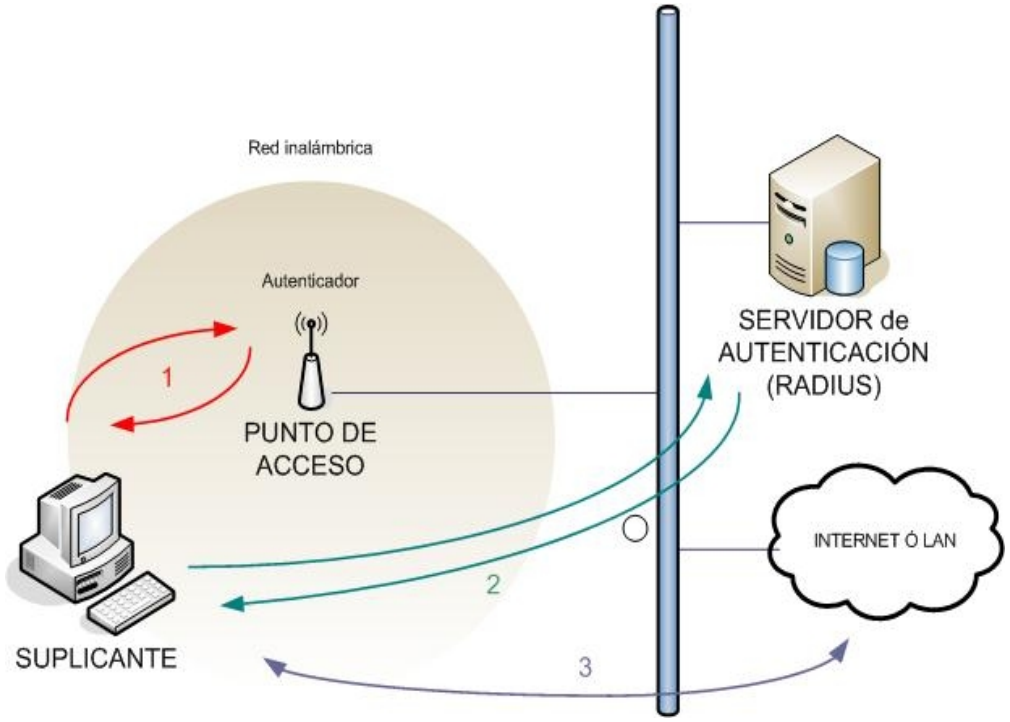
2.2.4 Control de Acceso a la red interna a proteger

2.2.4.1 El protocolo 802.1x

El protocolo de autenticación IEEE 802.1x (También conocido como Port-Based Network Access Control) es un entorno desarrollado originalmente para redes de cable, y posee mecanismos de autenticación, autorización y distribución de claves y además incorpora controles de acceso para los usuarios que se unan a la red. La arquitectua IEEE 802.1x está compuesta por tres entidades funcionales:

- El suplicante que se una a la red,
- El autenticador que hace el control de acceso,
- El servidor de autenticación que toma las decisiones de autorización.

En las redes inalámbricas, el punto de acceso sirve de autenticador. Cada puerto físico (puerto virtual en las redes inalámbricas) se divide en dos puertos lógicos, formando la PAE *(Port Access Entity)*. La PAE de autenticación siempre está abierta y permite el paso de procesos de autenticación, mientras que la PAE de servicio sólo se abre tras una autenticación existosa (por ejemplo, una autorización) por un tiempo limitado (3600 segundos por defecto). La decisión de permitir el acceso está hecha por lo general por la tercera entidad, el servidor de autenticación (que puede ser un servidor Radius dedicado o – por ejemplo las redes domésticas – un simple proceso funcionando en el punto de acceso).



El estándar 802.11i hace pequeñas modificaciones a IEEE 802.1x para que las redes inalámbricas estén protegidas frente al robo de identidades. La autenticación de mensajes se ha incorporado para asegurarse de que tanto suplicante como el autenticador calculan sus claves secretas y activan la enciptación antes de acceder a la red.

El suplicante y el autenticador se comunican mediante un protocolo basado en EAP. El rol del autenticador es, esencialmente, pasivo – se limita a enviar todos los mensajes al servidor. EAP es un entorno para el transporte de varios métodos de autenticación y permite sólo un número limitado de mensajes (Request, Response, Success, Failure), mientras que otros mensajes intermedios son dependientes del método seleccionado de autenticación: EAP-TLS, EAP-TTLS, PEAP, Kerberos V5, EAP-SIM, etc. Cuando se completa el proceso (por la multitud de métodos posibles no entraremos en detalles), ambas entidades (suplicante y servidor de autenticación) tendrán una clave maestra secreta. El protocolo utilizado en redes inalámbricas para transportar EAP se llama EAPOL (EAP Over LAN), las comunicaciones entre autenticador y servidor de autenticación utilizan protocolos de capa más alta, como Radius, etc.

2.2.4.2 Servidor de Autenticación RADIUS

Aunque en la especificación 802.1x se habla de los servidores de autenticación en términos genéricos, en la práctica se trata de elementos diseñados según los criterios del marco AAA (*Authentication, Authorization and Accounting*). Este marco define los elementos básicos necesarios para autenticar usuarios, manejar peticiones de autorización y realizar la contabilidad del sistema. Un servidor AAA debe ser capaz de recibir peticiones, examinar el contenido de dichas peticiones, determinar qué autorización se está pidiendo, recuperar las políticas que necesite de un repositorio, evaluar la petición y obtener la respuesta a la petición, o bien reenviar la petición a otro servidor AAA.

Autenticación	Verificar que una entidad es quien dice ser. Suele incluir unas credenciales (usuario/contraseña, certificados, tokens, etc.)
Autorización	Decidir si la entidad, una vez autenticada, tiene permiso para acceder al recurso.
Control de Acceso	Conceder el permiso definitivo. ACL. Registro, monitorización, contabilidad e informes, así como privilegios para el usuario.

RADIUS es un protocolo encuadrado dentro del marco AAA y utilizado principalmente en entornos donde los clientes (no confundir clientes de 802.1x con estaciones clientes) son elementos de acceso a la red (como los puntos de acceso). Estos elementos mandan información al servidor cuando una nueva estación cliente intenta conectarse, tras lo cual el servidor realiza el proceso de autenticación del usuario y devuelve al elemento de acceso la información de configuración necesaria para que éste trate al cliente de la manera adecuada. Toda la comunicación entre el elemento de acceso y RADIUS se encuentra cifrada mediante un secreto compartido que nunca se transmite por la red. Otro servidor de autenticación AAA es DIAMETER, el cual introduce algunas ventajas significativas respecto a RADIUS en materia de gestión de elementos de acceso complejos, si bien se encuentra aún en un estado menos avanzado de definición.

La mayoría de los Puntos de Acceso de hoy día permiten la integración con un servidor RADIUS. Además, éste puede integrarse con bases de datos externas: LDAP, Tacacs, NIS, PAM, Kerberos, NT Domain/Active Directory, otros servidores RADIUS ... para autenticar usuarios.

RADIUS envía mensajes utilizando el protocolo UDP (User Datagram Protocol). Utiliza el puerto 1812 para mensajes de autenticación y 1813 para mensajes de cuentas. En las RFCs 2865 y 2866 se definen los siguientes mensajes de RADIUS:

Access-Request	Lo envía un cliente para pedir una solicitud de autenticación y autorización al servidor para realizar una conexión.
Access-Accept	Enviado por RADIUS en respuesta a un Access-Request. Este mensaje informa al cliente RADIUS que el intento de conexión ha sido autenticado y autorizado.
Access-Reject	Enviado por RADIUS en respuesta a un Access-Request. Informa al cliente RADIUS que la conexión ha sido rechazada. El servidor RADIUS envia este mensaje si las credenciales son incorrectas o el usuario no está autorizado.
Access-Challenge	Enviado por RADIUS en respuesta a un Access-Request. Este mensaje es un reto que envia RADIUS al cliente y requiere una respuesta.
Accounting-Request	Enviado por un cliente para especificar información sobre su conexión.
Accounting-Response	Mensaje en respuesta a Accounting-Request.

Estos mensajes consisten en una cabecera RADIUS y cero o más atributos. Los atributos son datos sobre la petición de la conexión, como por ejemplo user name, user password, tipo de servicio, IP del servidor, etc...

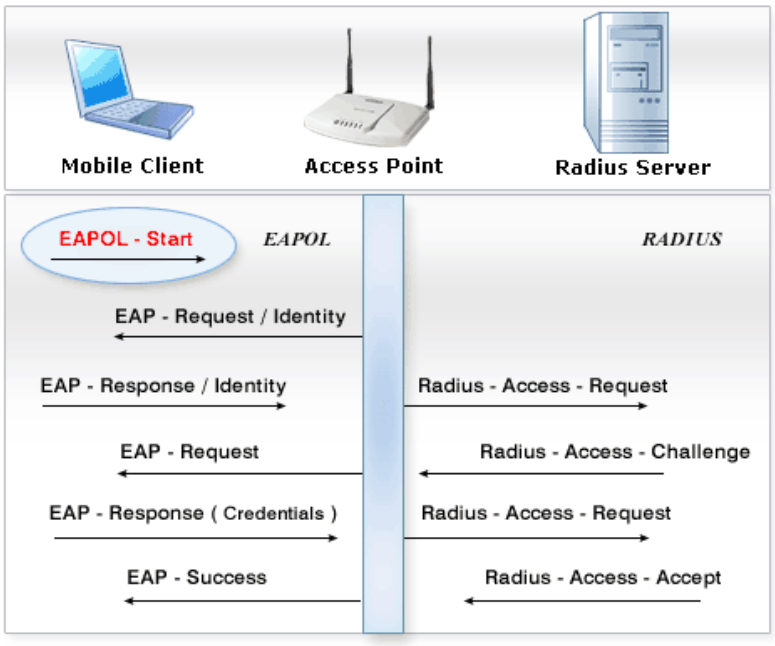
FreeRADIUS es una buena opción a la hora de escoger un servidor Radius. La versión 1.0 añade soporte para un gran número de EAPs y muy especialmente para PEAP. Los desarrolladores han introducido una opción para autenticarse en los dominios de Windows. Y Freeradius puede recuperar datos de cuenta desde fuentes típicas /etc/passwd, LDAP, MySQL, PostgreSQL o bases de datos Oracle.

2.2.4.3 Protocolo de autenticación EAP

Cuando un nodo inalámbrico pide acceso a la LAN, el punto de acceso pregunta por su identidad. Al nodo inalámbrico que pide autenticarse se le suele llamar *Suplicante*, el suplicante es el responsable de intercambiar datos de sus credenciales con el autenticador.

Como hemos comentado en la introducción de este capítulo, **EAP (Extensible Authentication Protocol)** es el protocolo utilizado para autenticarse, definido en la RFC 2284. Está basado en el protocolo PPP y proporciona un marco generalizado para diversos métodos de autenticación. EAP sirve como soprote a protocolos propietarios de autenticación, gestiona las contraseñas en mecanismos de desafío-respuesta y es capaz de trabajar con tecnología de clave pública.

Ejemplo de EAP:



1. El authenticator envía un paquete de "**EAP-Request/Identity**" al supplicant tan pronto como se detecte la asociación.

2. El supplicant envía un paquete de "**EAP-Response/Identity**" al authenticator, que pasa directamente al servidor de autenticación.

3. El servidor de autenticación envía un desafío al authenticator. El authenticator desempaqueta el contenido del paquete IP, lo empaqueta de nuevo en EAPOL y lo envía al supplicant.

4. El supplicant responde al desafío vía el authenticator y pasa la respuesta al servidor de autenticación.

5. Si el supplicant proporciona identidad apropiada, el servidor de autenticación responde con un mensaje de éxito al authenticator, que es pasado a sí mismo al supplicant. El authenticator permite a partir de este momento el acceso al supplicant.

Variantes de EAP: EAP define una variedad de métodos de autenticación. Algunos de los más importantes son:

- **EAP/MD5** transfiere un hash con el nombre del usuario, su contraseña y una cadena arbitraria. El servidor utiliza la clave en texto claro y la cadena arbitraria para generar su propio hash, el cual se compara con la hash entrante. Este método es simple, pero no es seguro contra ataques de tipo diccionario. Además, en una Wireless LAN, es imposible crear claves WEP dinámicas utilizando EAP/MD5. Por tanto, este método sólo está indicado para las pequeñas redes cableadas.

- **EAP/TLS**, tanto el servidor como el cliente necesitan certificados X.509. Este método es muy seguro, pero implica tener una PKI (Public Key Infraestructure) en funcionamiento.

- **PEAP**, (Protected Extensible Autentication Protocol). Con PEAP, sólo el servidor necesita un certificado para establecer una conexión TLS y enviar el nombre de usuario y la contraseña encriptados (MSCHAPv2, Microsoft Challenge Handshake Authentication Protocol). Los administradores sólo necesitan instalar el certificado del servidor en cada cliente. Cuando los clientes salen del sistema o cierran la conexión, PEAP detecta el cambio y finaliza la autorización, cerrando las conexiones por ambos lados.

En redes sólo cableadas, EAP/MD5 es a menudo la mejor opción. Esto es todo lo que se necesita para asignar dinámicamente VLANs y, a diferencia de PEAP, es un protocolo soportado por una gran cantidad de switches. Además de esto, el complicado esfuerzo administrativo es mucho menor que con PEAP o EAP/TLS.

Un switch normalmente proporciona funcionalidad NAS, traduciendo el protocolo EAPOL (EAP sobre LAN) desde el suplicante a Radius, que es lo que el servidor de acceso espera. La mayoría de los dispositivos ofrecen esta opción cuando se configura 802.1X. Necesitamos introducir la dirección y la clave para el servidor Radius. En muchos casos, los administradores pueden configurar múltiples servidores para proporcionar altos niveles de disponibilidad y ofrecer una solución alternativa en caso de que el servidor principal caiga.

Certificados Digitales: Un certificado se ofrece para garantizar una comunicación segura con el interlocutor que lo toma. Corre de cuenta del interlocutor verificar que el certificado que te ofrecen es correcto. Por ejemplo el servidor RADIUS ofrece un certificado para cifrar las comunicaciones entre él y el cliente que quiere acceder a la red. Cuando el cliente recibe el certificado, comprueba en una lista que tiene si viene firmado por una autoridad de confianza. En caso de que la firma del certificado esté avalada por una autoridad de confianza el cliente continúa normalmente su diálogo cifrando la información. Si la firma no está avalada por ninguna de las autoridades reconocidas, entonces avisa al cliente de que la firma no es de confianza y pregunta qué hacer. Es importante observar que la confianza o no de un certificado, en general, no impide que se pueda utilizar para establecer una comunicación, simplemente plantea dudas sobre la autenticidad del emisor.

Existen diversos formatos de certificados, pero el más extendido es el **X.509**, definido por la norma del ITU-T X.509. Un certificado X.509 es normalmente un fichero pequeño que contiene la siguiente información:

- Nombre distintivo del propietario.
- Nombre distintivo de la Autoridad certificadora. Identificación y firma de la Autoridad Certificadora (CA) que firmó el certificado.
- Período de vigencia. El período de tiempo de vigencia del certificado.
- Información adicional sobre la CA, números de serie o versión.

Una **Autoridad Certificadora (CA, Certificate Authority)** es la autoridad encargada de firmar los certificados y posteriormente confirmar que el dueño de un certificado es realmente quien dice ser. Para la firma de certificados, una Autoridad Certificadora puede establecer una política que especifique qué campos Nombres Distintivo es necesario incluir y cuáles son opcionales, y qué valores de los campos son o no admisibles.

2.2.4.4 Instalación y Configuración de FreeRADIUS en Debian

Por defecto, en Debian FreeRADIUS no viene compilado con el soporte para EAP+TLS debido a un problema de compatibilidad de la licencia OpenSSL y la licencia GPL. El requerimiento de la licencia OpenSSL especifica que hay que incluir un mensaje de agradecimiento al proyecto OpenSSL en el software donde se utilice. Parece que algunos proyectos han tenido que portar su código para el uso de GNU TLS para evitar este problema. OpenSSL es un paquete de herramientas de administración y librerías relacionadas con la criptografía, que suministran funciones criptográficas a otros paquetes como OpenSSH y navegadores web (para acceso seguro a sitios HTTPS). Estas herramientas ayudan al sistema a implementar el Secure Sockets Layer (SSL), así como otros protocolos relacionados con la seguridad, como el Transport Layer Security (TLS).

Para instalar FreeRADIUS nos descargaremos el código fuente desde la página oficial de FreeRADIUS, y compilaremos como mostramos a continuación:

```
cd /usr/src/
wget ftp://ftp.freeradius.org/pub/radius/freeradius-1.1.5.tar.gz
tar -xvfz freeradius-1.1.5.tar.gz
./configure
make & make install
```

Requisito indispensable es tener el compilador gcc y sus dependencias instalados en nuestra máquina:

```
apt-get install gcc
```

Llegado a este punto tendremos FreeRADIUS instalado, pasaremos por tanto a editar sus parámetros de configuración. Los archivos de configuración los encontraremos en */etc/freeradius/*

Los más destacables son los siguientes:

radiusd.conf: Archivo general de configuración de FreeRADIUS. En él podemos especificar parámetros como los directorios donde se encuentran los distintos ficheros que forman FreeRADIUS, el usuario y grupo con el que será ejecutado, el puerto donde escucha el servicio y si admite peticiones de cualquier dirección IP o sólo de aquellas que le especifiquemos, configuración de módulos, etc... La configuración de las variables se define de la siguiente forma: *variable = ${valor}*.

eap.conf: Archivo de configuración de las directivas EAP a utilizar. Es un *include* de radiusd.conf. Dentro de la sección definida por *eap { ... }* hay parámetros que configuran el tipo de protocolo EAP utilizado en la comunicación, el más importante es *default_eap_type*, donde especificamos si se utiliza eap, peap, etc.. Dentro de la sección *tls { ... }* definimos las claves y los certificados utilizados para encriptar la comunicación, tenemos variables como: *private_key_password, private_key_file, certificate_file*, etc...

clients.conf: Descripción y credenciales de los diferentes dispositivos que consultan al RADIUS. Se definen con el formato que indicamos a continuación, aunque es posible definir más parámetros que dan más información sobre el cliente:

```
cliente IP_dirección_cliente {
        secret = password
        shortname = nombre_cliente
}
```

users: Archivo donde se especifican las credenciales de los usuarios de la red. Se usa este archivo si no existe otro *backend* para el almacenamiento de los usuarios. (Como alguna base de datos externa o similar). El fichero define un primer campo que es el nombre del usuario seguido de la lista de requisitos de autenticación, como pueden ser un password, tipo de protocolo, dirección IP, mensaje de respuesta, etc...
La manera más simple de definir un usuario sería tal que así:

```
fran Auth-Type := EAP , User-Password == “secretfran”
```

Además, dependiendo del protocolo de autenticación que usemos, necesitaremos certificados digitales. La gestión de certificados se realiza mediante openssl.

```
apt-get install openssl
```

A continuación a modo de ejemplo crearemos una CA y un certificado digital válido para dicha CA. El fichero de configuración se denomina *openssl.cnf* y se encuentra en */etc/ssl.* Para localizar el fichero podemos ejecutar:

```
find -name openssl.cnf
```

Los certificados digitales incluyen cierta información que es necesario personalizar y adaptarla a nuestros datos. También se pueden modificar las rutas de la CA y de los ficheros de llaves, etc. En principio sólo nos interesa modificar los datos predeterminados de la entidad para definir los datos que aparecerán en los certificados.

Por ejemplo:

```
countryName_default = ES
```

Ahora crearemos una entidad certificadora (CA). Utilizaremos el directorio */etc/ssl/freeradius*

```
cd /etc/ssl/freeradius
mkdir private
openssl req -new -x509 -days 365 -newkey rsa:1024 -keyout /private/cakey.pem -out cacert.pem
chmod 600 /private/cakey.pem
```

Y genera toda la estructura de directorio necesaria, la llave privada (*cakey.pem)* y el certificado (cacert.pem). Podemos elegir un directorio cualquiera en lugar del propuesto.

Durante el proceso de creación de la CA tendremos que proporcionar la contraseña de la llave privada que tendremos que usar para crear certificados. Los nuevos certificados se crearán en */etc/freeradius/newcerts.*

El fichero index.txt lleva un log de todos los certificados validados por la autoridad certificadora:

```
mkdir newcerts
touch /etc/ssl/freeradius/index.txt
```

El archivo de texto serial lleva la cuenta del siguiente certificado X.509 disponible:

```
echo 01 > /etc/ssl/freeradius/serial
```

Hasta aquí ya tenemos creada la autoridad certificadora. Ahora crearemos un certificado y su corresopndiente clave privada:

```
openssl req -new -days 365 -newkey rsa:1024 -keyout sslkey.pem -out sslcert.pem
```

La clave privada se escribirá en *sslkey.pem* y la pública, todavía sin validar, en *sslcert.pem*

Por último, validaremos el certificado con el siguiente comando:

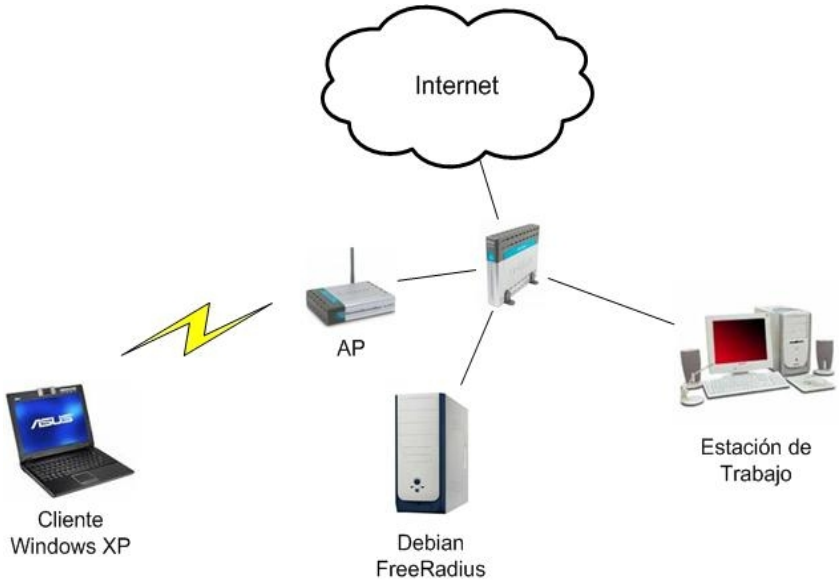
```
openssl ca -in sslcert.pem -out cert.pem
```

Ya podríamos borrar el certificado sin validar y quedarnos con cert.pem.

Hasta aquí el ejemplo de creación de certificados que podemos usar con nuestro FreeRADIUS.

2.2.4.5 Configuración de un cliente 802.1x en Windows XP y Linux

Con el objetivo de poner en práctica los conocimientos adquiridos sobre 802.1x realizaremos una batería de pruebas sobre el siguiente escenario real:



Tenemos un equipo portátil que es el cliente que desea acceder a la red destino: Internet. Disponemos también de un punto de acceso D-Link con soporte 802.1x que es el elemento que actúa como autenticador. Éste se conecta físicamente mediante cable Ethernet con un Router-ADSL y que a su vez conecta con el servidor FreeRADIUS con nuestro dispositivo Debian. Por último, disponemos de una estación de trabajo para configuración y monitorización remota de los elementos de red y del servidor.

Utilizaremos como suplicantes soporte nativo de Windows XP en primer lugar, y wpa_supplicant en Ubuntu en segundo lugar.

Tendremos que usar en esta sección algunos conceptos como *direcciones de red*, *puertos* y *protocolo TCP/IP*, sin embargo tranquilizaremos al lector anunciándole que introduciremos detenidamente estos conceptos en el siguiente capítulo.

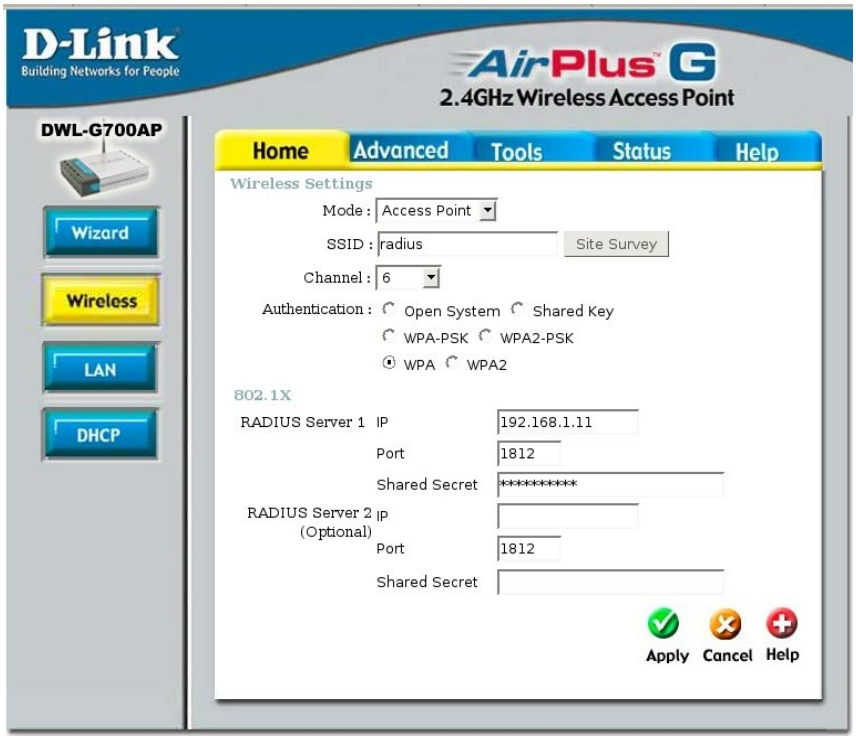
En primer lugar editamos el archivo */etc/freeradius/clients.conf* de nuestra máquina Debian y añadiremos unas líneas de texto para habillitar que nuestro punto de acceso se comunique con él.

```
client 192.168.1.50 {
        secret = testing123
        shortname = DWL-AP
}
```

Donde *192.168.1.50* es la dirección de red que tiene el punto de acceso, *secret* es la clave compartida para encriptar los paquetes entre el punto de acceso y FreeRADIUS y *shortname* es utilizado como un alias.

Utilizando la estación de trabajo configuraremos el punto de acceso mediante su interfaz WEB, para habilitar 802.1x. Para ello seleccionamos 802.1x como método de *Authentication,* la dirección IP de red de nuestra máquina Debian para que mande las peticiones de autenticación al servidor de autenticación, y el puerto en el que escucha (1812 por defecto). Además, la clave secret anterior, *testing123*.

A continuación presentamos una captura del interfaz de configuración del punto de acceso:



En nuestras pruebas utilizaremos el protocolo PEAP. El proceso de autenticación de PEAP consta de dos fases principales:

1. Autenticación de servidor y creación de un canal de cifrado TLS. El servidor se identifica ante un cliente proporcionando a éste información del certificado. Una vez que el cliente comprueba la identidad del servidor, se genera un password maestro. A continuación, las claves de sesión obtenidas del password maestro se utilizan para crear un canal de cifrado de TLS que cifra toda la comunicación posterior entre el servidor y el cliente inalámbrico.
2. Conversación de EAP y autenticación de usuario y de equipo cliente. Mediante el canal de cifrado de TLS se encapsula una conversación de EAP completa entre el cliente y el servidor. Con PEAP puede utilizarse cualquiera de los diversos métodos de autenticación de EAP, como contraseñas, tarjetas inteligentes y certificados, para autenticar al usuario y al equipo cliente.

Para configurar PEAP en el servidor editaremos el archivo eap.conf de la siguiente manera:

```
eap {
        default_eap_type = peap
        ....
}
```

Además de descomentar las líneas de peap y tls.
Utilizaremos para el servidor el certificado generado por defecto:

```
tls {
        private_key_password = whatever
        private_key_file = ${raddbdir}/certs/cert-srv.pem
        certificate_file = ${raddbdir}/certs/cert-srv.pem
        CA_file = ${raddbdir}/certs/demoCA/cacert.pem
        dh_file = ${raddbdir}/certs/dh
        random_file = ${raddbdir}/certs/random
        ...
}
```

Donde la variable *raddbdir* toma el valor definido por defecto en *radiusd.conf* que es: */etc/freeradius*.

En el fichero *users* añadiremos las credenciales de un usuario de prueba:

```
fran Auth-Type := Local, User-Password == “secretfran”
        Reply-Message = “Hola, fran”
```

Para añadir otro usuario insertamos en una nueva línea el username y los requisitos de autenticación con la misma sintaxis que acabamos de ver. FreeRADIUS analiza el fichero *users* de arriba a abajo y cuando encuentra una entrada que coincide con el nombre de usuario, le aplica las características de autenticación que le siguen. Existe una entrada especial llamada DEFAULT, que es como un comodín que coincide con todos los nombres de los usuarios, de esta manera podríamos especificar condiciones de autenticación para muchos usuarios a la vez.

Algunos de los atributos que podemos especificar son:

Auth-Type: tipo de autenticación, ejemplo: local, eap...
Reply-Message: mensaje de repuesta al usuario.
User-Password: clave de autenticación.
Fall-Through: si vale 1 RADIUS seguirá chequeando el fichero users para ver si hay más coincidencias.
Framed-IP-Address: para especificar una dirección IP estática.

Inicialmente querremos ejecutar el servidor RADIUS en modo *debug:*

```
freeradius -X
```

Le dice al servidor que muestre por consola los mensajes de error y las advertencias.
El programa de la línea de comandos *radtest* proporciona una herramienta útil de comprobación de usuarios que puede ayudarnos a probar nuestro *users*. Suponiendo la configuración mencionada, la línea de comandos para comprobar nuestro usuario fran es la siguiente:

```
radtest fran secretfran localhost 0 testing123
```

Donde *fran* y *secretfran* es el username y la clave del usuario que desea autenticarse, *localhost* es la dirección donde está el servidor RADIUS (lo estamos probando sobre la misma máquina), 0 es el NAS-Port que realmente no importa qué valor tome, y *testing123* es la clave que le dimos a nuestro cliente (NAS, es decir, el punto de acceso inalámbrico). Esta simulación sirve para comprobar que cuando un usuario se autentica ante el sistema, FreeRADIUS le da permiso o no según queramos. Nuestra respuesta es:

```
rad_recv: Access-Accept packet from host 127.0.0.1:1812, id=7, length=32
        Reply-Message = “Hola, fran”
```

Ahora cambiaremos el método de autenticación de nuestro usuario de prueba a EAP:

```
fran Auth-Type := EAP, User-Password == “secretfran”
```

Una vez hecho esto, tenemos nuestro punto de acceso listo para autenticar usuarios. Es hora de que el usuario que está sentado ante su portátil introduzca sus credenciales y pueda conectarse a la red, para ello necesitamos un 802.1x supplicant.

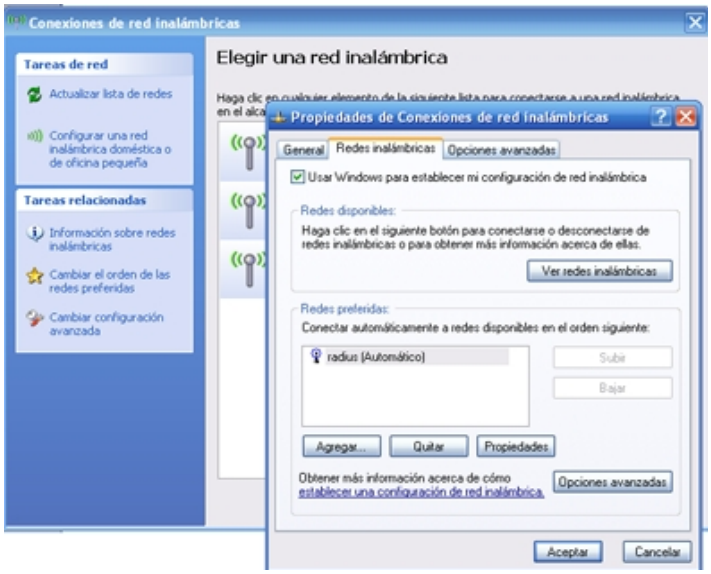
En la siguiente tabla mostramos clientes de 802.1x para distintos sistemas operativos:

Windows	Soporte nativo del sistema Windows XP SP2 AEGIS Client: http://www.mtghouse.com
Linux	Xsupplicant: http://www.open1x.org/ AEGIS Client: http://www.mtghouse.com wpa_supplicant: http://hostap.epitest.fi/wpa_supplicant
Mac OS X	Soporte nativo del sistema AEGIS Client http://www.mtghouse.com

Utilizando como suplicante soporte nativo de Windows XP:

Desde el menú de conexiones de red inalámbricas *clikeamos* en “Cambiar configuración avanzada”, despues de eso en Redes preferidas tenemos el perfil “radius”, que es información almacenada sobre nuestra red inalámbrica.

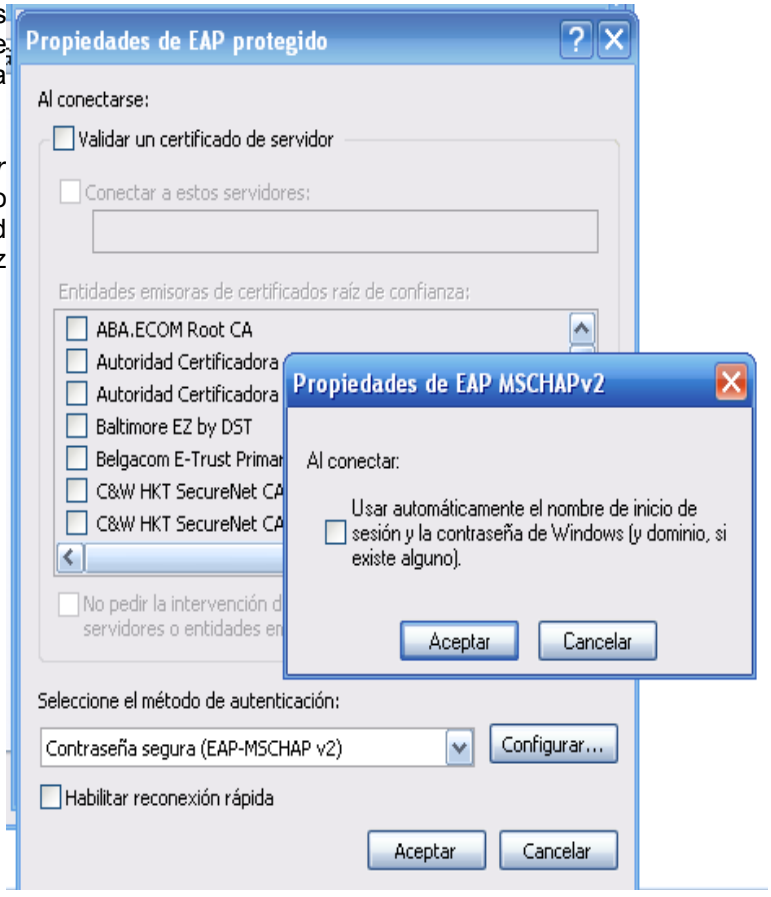
Pulsando en el botón Propiedades para la red *radius* nos aparece la siguiente pantalla:

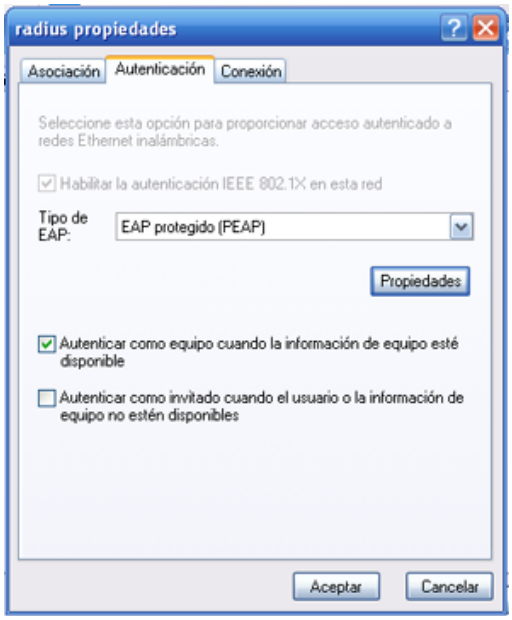


Seleccionamos como “Tipo de EAP”: EAP Protegido (PEAP).

A continuación seleccionamos *Propiedades*.Nos aparece la ventana de Propiedades de EAP que observamos a nuestra derecha:

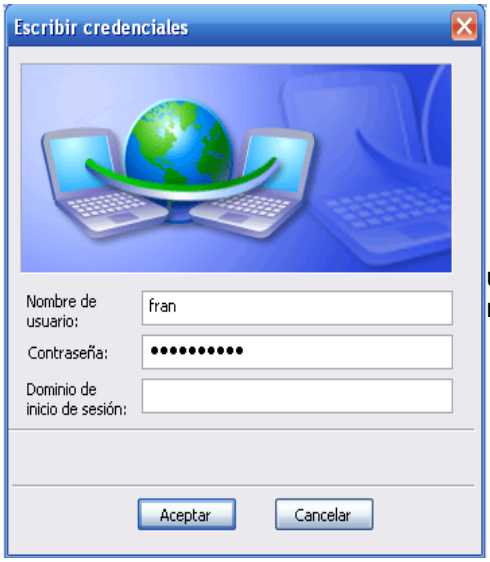
Deseleccionamos la casilla *Validar Certificado de servidor.* Ya que nuestro certificado está generado por una Autoridad Certificadora propia, por tanto no tiene validez en la lista de servidores que viene por defecto.



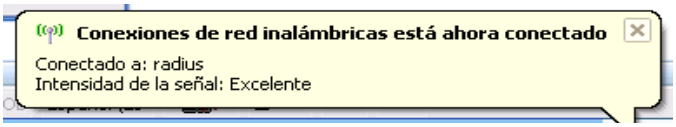


Por último clickeamos en “Configurar...” y sobre la nueva ventana deseleccionamos la opción de usar automáticamente el nombre de inicio de sesión y la contraseña de Windows. De esta forma, se nos preguntarán nuestras credenciales cuando tengamos conexión a Radius para validar nuestra sesión.

Al intentar realizar la conexión nos aparece:



Y finalmente después de introducir nuestro nombre de usuario y contraseña (aquél que definimos en el fichero *users*) nos aparece:



Utilizando como suplicante wpa_supplicant en Linux:

En este caso basta con editar la configuración del archivo /etc/wpa_supplicant.conf a:

```
ctrl_interface =/var/run/wpa_supplicant

network ={
        ssid = “radius”
        proto = WPA RSN
        key_mgmt = WPA-EAP
        pairwise = TKIP CCMP
        group = TKIP CCMP
        eap = PEAP
        identity = “fran”
        password = “secretfran”
}
```

La variable *ctrl_interface* especifica una ubicación llamada interfaz de control que está disponible para que programas externos se comuniquen e interactúen con wpa_supplicant, básicamente es un directorio donde se crean sockets de comunicación.

Network es un bloque que engloba la configuración para un punto de acceso. El fichero wpa_supplicant.conf puede tener varios bloques network en orden de preferencia de arriba a abajo, y se utilizará el primero que coincida con el ssid de la red a la que queremos conectarnos.

Aunque se pueden especificar más parámetros dentro de este bloque, los que hemos usado son:

ssid (obligatorio): el nombre de la red.
proto: lista de protocolos aceptados, puede tomar los valores:
WPA = WPA/IEEE 802.11i/D3.0
RSN = WPA2/IEEE 802.11i).
Por defecto vale WPA RSN.
key_mgmt: lista de protocolos de gestión de claves. Puede tomar los valores: WPA-PSK. WPA-EAP, IEEE8021X o nada.
En nuestro caso hemos elegido WPA-EAP.

pairwise: tipo de algoritmo de cifrado utilizado en comunicaciones unicast en WPA.
Puede ser:
CCMP = AES con CBC-MAC (RFC 3610, IEEE 802.11i/D7.0)
TKIP = Temporal Key Integrity Protocol (IEEE 802.11i/D7.0)
o ninguno.
Por defecto es TKIP CCMP.
group: Cifrado utilizado en broadcast en WPA. Similar al parámetro anterior.
eap: lista de métodos EAP aceptados. Pueden ser: MD5, MSCHAPv2, TLS, PEAP, TTLS ...
Nosotros elegimos PEAP.
identity: usuario para la autenticación.
password: contraseña.

Ahora levantamos la interfaz de red inalámbrica con la nueva configuración:

```
ifup eth1
```

Al mismo tiempo podemos observar en la consola de radius el intercambio de mensajes de autenticación. Mostramos a continuación uno de ellos (el Access-Accept enviado desde el servidor al cliente).

```
Sending Access-Accept of id 43 to 192.168.1.50 port 3072
    Framed-IP-Address = 255.255.255.254
    Framed-MTU = 576
    Service-Type = Framed-User
    Reply-Message = "Hola, fran"
    MS-MPPE-Recv-Key =
0x146a5bf4e7459211b49a0b30c412fcdf39825486d704dda7886bdd45730c2257
    MS-MPPE-Send-Key =
0x4201c4fda81a292a041ed158ccfee8d86d4efbcde127c78a3399eec8a6cf1e23
    EAP-Message = 0x03080004
    Message-Authenticator = 0x00000000000000000000000000000000
    User-Name = "fran"
```

Hecho esto, ya tenemos acceso a la red y por tanto, podremos navegar por Internet desde nuestro portátil.