

Inalámbrico COMO

Publicado en

<http://www.marianistas.org/anavegantes>

Roberto Arcomano berto@fatamorgana.com

Traducido por:

Montse Herrera montse@redlibre.net

Jaime Robles jaime@redlibre.net

Alejandro Pallarés alejandropallanet.com

v1.3, 12 de Noviembre de 2001 - Traducción v1.0, 21 Noviembre de 2001

La inalámbrica es una nueva tecnología de tarjetas de red, con alta velocidad (hasta 11 Mbps). Este documento explica como configurar una red inalámbrica en Linux, problemas de compatibilidad, algunos requerimientos geográficos y más. La última versión de este documento puedes encontrarla en <http://www.fatamorgana.com/bertolinux>

1. Introducción

1.1 Introducción

Este documento explica algunos aspectos sobre Redes Inalámbricas, como configurarlas, y posibles problemas que puedan surgir. A diferencia de la red de cable, la red inalámbrica requiere algunos trucos para que funcione correctamente. Debes saber algo sobre antenas, como orientarlas, itinerancia y algunos aspectos más. Acepto correcciones. Puedes encontrar más ayuda en [Jean Tourrilhes Wireless Howto](#)

Para alguna sugerencia o corrección escribe a mi [dirección de correo](#)

1.2 Copyright

Copyright (C) 2000,2001 Roberto Arcomano.

Esto documento es libre; puedes redistribuirlo y/o modificarlo bajo los términos de la Licencia Pública General GNU como se publica por la Fundación para el Software Libre; o por la versión 2 o (a tu elección) posterior de esta licencia.

Este documento se distribuye con la esperanza de que sea útil, pero SIN GARANTÍA ALGUNA; incluso sin la implícita de MERCANTILISMO o MANTENIMIENTO PARA PROPÓSITO PARTICULAR. Ver la Licencia Pública General GNU para más detalles. Puedes conseguir una copia de ella en [aquí](#)

1.3 Traducciones

Si quieres traducir este documento eres libre de hacerlo, sólo tienes que:

1. Comprobar que no existe otra versión en tu LDP (Proyecto de Documentación Linux) local
2. Mantener toda la sección 'Introducción' (incluidas 'Introducción', 'Copyright', 'Traducciones', 'Créditos').

¡Gracias por tu traducción!

1.4 Créditos

Gracias a [Fatamorgana Computers](#) por el equipamiento hardware y la oportunidad de experimentar.

Gracias a [Linux Documentation Project](#) por publicar y subir mi documento muy rápidamente.

2. Conocimientos Básicos

2.1 ¿Qué es Inalámbrico?

La Inalámbrica es una nueva tecnología que te puede ayudar a conectar ordenadores a distancia. Funciona con tarjetas inalámbricas con un TX/RX interno a 2.4 GHz mientras que la interfaz de software es del tipo ethernet, con una dirección física diferente para cada tarjeta en el mundo. Normalmente la potencia de transmisión es de 10-20 mW hasta 100mW (ver estándar IEEE 802.11 y licencias FCC/CEPT).

2.2 ¿Cuál es la distancia máxima entre tarjetas de radio?

El aspecto más importante en la comunicación inalámbrica es una línea visual clara: DEBES VER (a simple vista o con binoculares) una antena desde el otro extremo pudiendo existir entre ambos un pequeño árbol como máximo.

La distancia depende de la antena (y eventualmente de un amplificador) utilizada: de 2 a 300 metros con una antena omnidireccional; 1 km con una direccional; de 2 a 3 km con una omnidireccional amplificada(200mW); algunos kms. con una antena parabólica. 50 a 60 kms con una antena parabólica o direccional amplificada (algunos watos).

Ten en cuenta que no siempre el uso de tarjetas inalámbricas amplificadas es legal, ya que puedes violar las especificaciones FCC/CEPT (y también las propias de tu país).

2.3 ¿Qué diferencia hay entre una red de cable y una inalámbrica?

Las redes de cable son muy sencillas de configurar (al menos a bajo nivel). Las inalámbricas son muy difíciles de configurar, de administrar, y depurar... El típico problema con una red de cable es la instalación del hardware, del software, la depuración y demás, que pueden llegar a ser muy críticos en una red inalámbrica:

1. Tienes que elegir la tarjeta inalámbrica correcta: existen muchas tarjetas de diversos fabricantes con requerimientos y especificaciones diferentes. Si quieres crear una pequeña LAN/WAN tienes que comprar tarjetas que cumplan el estándar IEEE 802.11 con un Punto de Acceso.
2. Muchas tarjetas son PCMCIA, por tanto primero tienes que instalar las fuentes pcmcia de Linux.
3. Debes comprobarlo con dos sistemas funcionando, primero a corta distancia, para que luego puedas alejarlos.
4. Debes probarlo con distintas condiciones atmosféricas (normalmente con lluvia).
5. Por último felicitarte por haberlo hecho funcionar.

¡Si instalas un repetidor(Caja LiNIX con varias tarjetas de red de cable e inalámbricas) puedes tener problemas para modificar su configuración a distancia!

2.4 ¿Qué necesito saber para configurar una red inalámbrica?

Existen una serie de requisitos para configurar una red inalámbrica;

Requerimientos software:

1. Conceptos generales como dirección IP, máscara de red, enrutado... incluidos en el Linux NET3-4-HOWTO; *
2. Conceptos específicos de red como proxy arp, bridging, proc fs, contenidos en Proxy-ARP-Subnet, Bridge Mini-Howto y en las fuentes del kernel LiNIX (2.2.x o 2.4.x) en Documentation/networking/ ip-sysctl.txt) *
3. Conocimientos de red inalámbrica como modo de acceso (ADHOC, INFRASTRUCTURA y PUNTO DE ACCESO), concepto de canal, definiciones de exterior e interior y algunos más que puedes encontrar en algunos documentos que tratan sobre redes inalámbricas: estándar IEEE 802.11, CEPT, etc.

requerimientos no-software:

1. Experiencia mínima en antenas, montaje y orientación
2. Instalación de hardware de PC con especial cuidado en no producir interferencias entre diferentes tarjetas inalámbricas (si es necesario).

¡Finalmente mucha suerte!

* Todos los COMOs necesarios para crear este documento pueden obtenerse en <http://www.linuxdoc.org/>

2.5 ¿Porque debo montar una red inalámbrica y que puedo esperar de ella?

¿Por qué? ¿Porque no estás satisfecho con una red de cable!

Con tarjetas inalámbricas puedes cruzar jardines, parques, casas, (¡pero DEBES VER el otro extremo!).

El protocolo de Alto Nivel usado en las tarjetas inalámbricas es el mismo que en las tarjetas ethernet: TCP/IP sobre ethernet inalámbrica pero debes prestar atención al compartir aplicaciones Windows, ya que si usas Linux para reenvío, debes tener cuidado con un repetidor IP ya que no permite pasar mensajes broadcast (ver más sobre el protocolo NetBIOS): en este caso debes usar un servidor WINS para soportar Examinador de Red (ver documento sobre Samba).

La tecnología inalámbrica te permite crear una pequeña LAN/WAN con un punto de acceso central (¡incluso con acceso a Internet!) y dar acceso a cualquiera por aire.

Imagina todo un país conectado por máquinas de radio.

Imagina una red que puede conectar a toda la gente de un país, compartiendo archivos, aplicaciones de audio, de vídeo con un amplio ancho de banda (como una red de cable).

Todo lo que puedes hacer (que ya está funcionando en algunos países) usando tarjetas inalámbricas, puntos de acceso y cajas Linux, que funcionan como repetidor (a nivel IP como un router o, si quieres, a nivel de enlace de datos, como bridge, ver detalles en [Bridge Http Link](#) o [Bridge Ftp Link](#)

2.6 ¿Qué tarjetas inalámbricas están incluidas en este COMO?

En este COMO empleo una configuración genérica (como introducción a las redes inalámbricas), y describo un ejemplo para cada tarjeta que he probado Personalmente, con algunos trucos que pueden mejorar su rendimiento.

Lista de tarjetas inalámbricas:

1. Proxim Symphony - <http://www.proxim.com/>
2. Webgear AviatorPRO 2.4 (necesita soporte pcmcia) - <http://www.webgear.com/>
3. Lucent Wavelan I, II, Orinoco - <http://www.lucent.com/> and <http://www.orinoco.net/>
4. Cabletron - <http://www.cabletron.com/>
5. YDI am930_isa - <http://www.ydi.com/>
6. Siemens Radio Modem (Dect) - <http://www.siemens.com/>
7. RadioLan (5 GHZ) - <http://www.radiolan.com/>

Para una lista más exhaustiva ver [Jean Tourrilhes Wireless Howto](#).

El Siemens Radio Modem no es realmente una tarjeta inalámbrica 802.11, es un modem que se conecta al puerto de serie y actúan como módem (a 1800 MHz, con tecnología DECT). Su uso se describe en el Apéndice B.

Las tarjetas RadioLan funcionan a 5.4GHz en un entorno Windows 9x y con un punto de acceso RadioLan que enlaza entre redes de cable e inalámbricas (no existe driver conocido para Linux).

2.7 ¿Cuánto cuestan?

Las tarjetas que aparecen más abajo tienen un coste muy bajo: desde algunos cientos de dólares hasta algunos miles de dólares para los puntos de acceso (incluyen 2 tarjetas inalámbricas (Lucent, por ejemplo) que pueden actuar como bridge.

3. Información técnica acerca de la tecnología inalámbrica.

Aquí expongo alguna información técnica para entender las bases de los entornos inalámbricos.

3.1 Capa física

En la primera capa ISO/OSI podemos tener 3 tipos de aspectos:

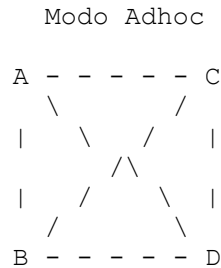
1. FHSS, Agilidad de frecuencia de espectro expandido (Frequency Hopping Spread Spectrum)
2. DSSS, Secuencia directa de espectro expandido (Direct Sequence Spread Spectrum)
3. Conexiones infrarrojas, no cubiertas en este COMO

3.2 Configuraciones

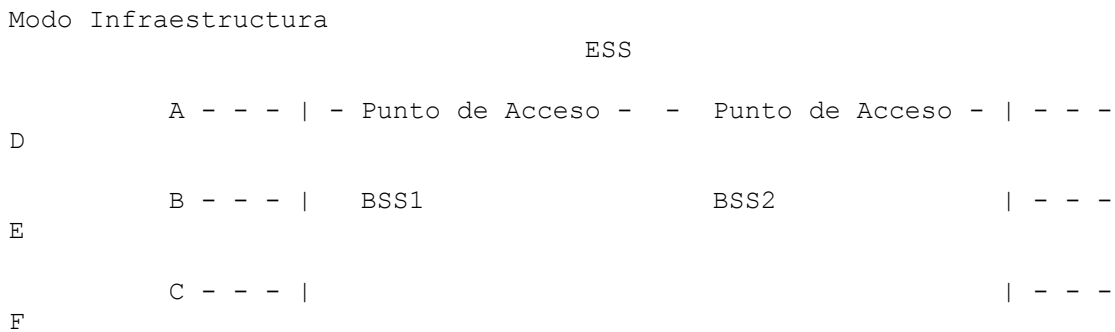
2 tipos de configuraciones:

1. Modo AdHoc (también llamado modo independiente), donde hay redes independientes con un BSS (Conjunto de Servicio Básico) cada una. Cada estación tiene el mismo BSS.
2. Modo infraestructura, donde el número de redes (con un BSS cada uno) puede comunicarse unas con otras gracias a un Punto de Acceso (uno por cada BSS) para crear un ESS (Conjunto de Servicio Extendido). También tiene la función de itinerancia que permite a una estación "engancharse" al Punto de Acceso más cercano.

Adhoc es el método más sencillo (y también el menos escalable) y permite a muchas máquinas comunicarse con otras directamente. El requisito restrictivo es que todas deben ser visibles entre ellas para llegar a una cobertura completa en la red. (¡Al menos idealmente porque este problema puede ser solucionado en la capa IP! Para más información mirar Par 5.4).



En un entorno de Infraestructura puedes usar un Punto de Acceso al que TODAS las máquinas deben conectarse para compartir la red.



B y C no pueden ver a D, E y F, pero pueden comunicarse porque todas usan el mismo ESS. Importante: A,B y C podrían también no verse entre ellas.

Además hay términos como interior y exterior para distinguir un área de cobertura pequeña de un área grande de cobertura.

3.3 Compatibilidad

Tenga en mente que hay muchas tarjetas inalámbricas en el mundo, pero no todas las tarjetas pueden comunicarse con cualquier otra. Para hablar entre ellas deben usar:

1. misma configuración de modo: todas Adhoc o todas Infraestructura
2. la misma capa física: todas DSSS o todas FHSS
3. el mismo protocolo (por ejemplo Proxim tiene su protocolo propietario particular OpenAir y no puede hablar con otras tarjetas FHSS).

3.4 ¿Debo usar Adhoc o Infraestructura?

Un Punto de Acceso es muy útil y elimina problemas pero son caros. Idealmente, para una red más concentrada puede usar el modo Infraestructura, mientras que para pocas máquinas puede elegir Adhoc: ¿por qué gastar mucho dinero para pocas máquinas?

De cualquier forma sepa que si gasta mucho dinero posiblemente todo funcione bien mientras que si gasta poco tenga algún problema.

3.5 ¿No puede actuar un LINUX como un Punto de Acceso?

¡Buena pregunta! Desafortunadamente no hay ese tipo de programas (libres) que puedan hacerlo, así que tenemos que arreglárnosla con el modo Adhoc o comprar más Puntos de Acceso.

Nota: si usted es un experto en ingeniería inversa puede descargarse el firmware de un Punto de Acceso, usar un intérprete compatible con el procesador y hackear el código de un Punto de Acceso y reescribirlo para LINUX.

4. Herramientas requeridas

4.1 Hardware requerido

Necesitas un LINUX (486 o mejor un Pentium 100+ con 16MB+ de RAM), la tarjeta de red inalámbrica, una antena (mira la parte 2.2). ¡Necesitas lo mismo en el otro extremo (con Win9x o WinNT, si prefieres...) porque tienes que simular una comunicación!

4.2 Programas requeridos

Necesitas:

1. las fuentes de un kernel estable reciente (2.2.x)
2. fuentes del paquete pcmcia estable reciente (pcmcia-cs) si compras una tarjeta pcmcia
3. El driver de la tarjeta inalámbrica: si no lo tienes puedes descargarlo de la web del fabricante o del vendedor. Si no lo encuentras puedes buscar en el [Wireless-howto de Jean Tourrilhes](#).

¡¡Si incluso así no lo encuentras deberás esperar o convertir un driver de Windows en un driver para LINUX!! (¡buena suerte!).

Después de eso, deberás recompilar el núcleo, recompilar las fuentes pcmcia (si las necesitas para la tarjeta inalámbrica), finalmente recompilar el driver de la tarjeta. Esta es la situación general, puede ser que algunas tarjetas sólo requieran el paso 3 o 1 y 3, depende de cada driver específico.

5. Vistazo general a la configuración de una red inalámbrica.

5.1 Pasos fundamentales

Una vez que tienes el material necesario y que has compilado todo lo que necesitas, deberás hacer el paso fundamental en la configuración de un sistema inalámbrico:

1. Configuración de bajo nivel del kernel para permitirle ver tu tarjeta inalámbrica (a bajo nivel, los puertos IO, interrupciones, DMA, ...): debes ver la clase de mensajes que dicen que tu tarjeta ha sido encontrada y configurada correctamente.

2. Para la configuración del nivel de enlace para cada tarjeta existe una utilidad que puede poner el típico valor del nivel de enlace inalámbrico. Por ejemplo en Proxim Synphony la utilidad se llama "rl2cfg" mientras que en la configuración de tarjetas pcmcia la configuración se hace en los ficheros de pcmcia. Tienes que configurar todas tus tarjetas inalámbricas de forma coherente para que hablen entre ellas.
3. Configuración IP debes ser capaz de usar las funcionalidades de ifconfig y route para cambiar la configuración IP.
4. Trucos para un mejor rendimiento y para evitar conflictos. Ahora tu red inalámbrica está funcionando básicamente: además tienes que ajustar algunas configuraciones particulares como el proxy-arp, el icmp echo redirect, el bridging, el canal y ese tipo de cosas para optimizar la red y evitar conflictos extraños y comerse inútilmente el ancho de banda.

N.B.: los pasos 1, 2 y 3 se corresponden con los niveles 1, 2 y 3 del estandar ISO/OSI mientras el paso 4 es un añadido para solucionar la situación generada por la máscara de red 255.255.255.255. De hecho la máscara de 32 bits viola el estandar ISO/OSI porque fuerza que se use la misma dirección para el broadcast y la IP de la máquina y la dirección de red no existe.

Alguien puede criticar este punto de vista, pero si usas el estandar ISO/OSI para configurar una red inalámbrica perderás muchas direcciones configurando subredes; por cada subred habitualmente descartas dos direcciones IP (red y broadcast) y no puedes conseguir flexibilidad en la asignación de IP (geográficamente hablando). Puedes encontrar más información sobre ésto en el Apendice A.

Date cuenta de que el paso 2 no está presente en las tarjetas con hilos porque no hay configuraciones especiales para hacer esto.

5.2 Configuración a bajo nivel del kernel

Siempre es un problema para la administracion del PC: para el kernel (o en general) mira tu hardware.

Las tarjetas inalámbricas son más complejas porque muchas de ellas, habitualmente tienen un interfaz PCMCIA, así que primero de todo tienes que hacer que tu kernel vea el adaptador PCMCIA y después intentar instalar el driver específico del hardware de tu tarjeta inalámbrica.

Así que en la configuración PCMCIA tendrás que:

1. instalar las fuentes del kernel de linux de <http://www.kernel.org/> en /usr/src/linux (mira las utilidades tar y gzip)
2. instalar las fuentes PCMCIA de <ftp://projects.sourceforge.net/pub/pcmcia-cs> para instalarlo en /usr/src/pcmcia (mira tar y gzip)
3. configurar y recompilar tu kernel: mira el fichero README en el directorio del kernel (/usr/src/linux)
4. configurar y recompilar las fuentes PCMCIA: usa configure y make en /usr/src/pcmcia. Asegúrate de que el driver está ahí, si no tendrás que instalarlo y seguir las instrucciones (habitualmente un tar zxvf driver.tgz desde el directorio

pcmcia es suficiente). Después teclea "make all" para compilar. Para terminar teclea "make install".

5. Una vez tecleado install encontrarás ficheros de configuración en /etc/pcmcia.

<p>

En el caso no PCMCIA:

1. Si tu driver está presente (99% no) en las fuentes de linux, tienes que instalarlo en un directorio y compilarlo.

Una vez que sabes el nombre del módulo tienes que cargarlo: en la configuración PCMCIA sólo tienes que arrancar el demonio PCMCIA (/etc/rc.d/init.d/pcmcia start en RedHat), para otra "modprobe module_name options". Con las opciones le darás el puerto IO, IRQ, y configuración del enlace (mira Par 5.3) del driver inalámbrico. De todas formas las herramientas para saber si el hardware se ve correctamente con el driver son:

1. "tail /var/log/messages" que da información del syslog
2. "dmesg" para mayor información.
3. /proc dir: ioports, devices, irq ficheros y subdirectorios específicos del driver.

5.3 Configuración del nivel de enlace

¿Qué es eso?

Las redes de cable sólo necesitan conectarse unas a otras y se podrán configurar los parámetros de TCP/IP.

Por contra, las redes inalámbricas necesitan configuración de la capa de enlace como:

1. ¿A qué tipo de red inalámbrica pertenezco? (Adhoc o Infraestructura)
2. ¿Qué canal debo usar?
3. ¿A qué subred (BSSID) pertenezco y cuál es mi id ESS?
4. ¿Está mi comunicación protegida por un algoritmo de encriptación? ¿Cual es la longitud de clave?

Como ves hay muchas configuraciones que tienes que ajustar, la razón viene de la arquitectura de una red inalámbrica: puede haber alguien cerca que pueda ver tus paquetes, usar tus servicios únicamente apuntando su antena en la dirección adecuada y configurando bien los parámetros TCP/IP.

Además pueden existir muchas subredes inalámbricas que pueden generar interferencias unas a otras.

Así que hay:

1. Opciones para el módulo load-timee: "modprobe ray_cs essid='LINUX'" por ejemplo o
2. Utilidades para el driver run-time: "r12cfg eth1 master".

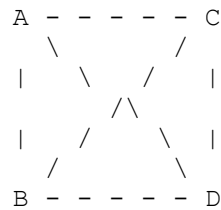
5.4 Configuración IP

Éste es el tercer problema al que hay que enfrentarse. Esta situación se vuelve problemática únicamente cuando tu red empieza a evolucionar hacia una red más grande.

¡Recuerda que las redes IP inalámbricas no te forzarán si no las fuerzas tú!

Una configuración sencilla

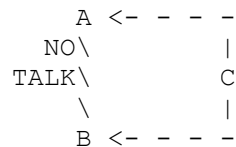
Todas las máquinas se ven entre ellas



Una configuración como ésta es muy sencilla y no requiere nada en especial (a nivel IP): sólo necesitas asignar una dirección IP para cada máquina y asignar una máscara de red global coherente.

Una configuración más compleja

A no ve directamente a B



Aquí A y B pueden comunicarse pasando a través de C.

Si la red es en modo Infraestructura y C es el Punto de Acceso todo es correcto. Si es en modo Adhoc, también puedes diseñar una máquina como "master" (¡sé que el término no es demasiado formal!), una máquina que crea un BSS al que la otra máquina puede unirse.

Ahora se alcanza conectividad plena en la capa IP: A y B pueden hablar con C usando el mismo interfaz de C, así que si haces ping desde A a B recibirás muchos paquetes ICMP REDIRECT de C porque C le está diciendo a A que el destino está en la red de la que llega la petición.

Solución: tecllea "echo 0 > /proc/sys/net/ipv4/conf/ethx/send_redirects" (donde ethx es el interfaz de C hacia A y C) para anular eso.

Otro problema: ¿Qué máscara asigno a A y C? Si asignas a A una máscara que incluya a A y C nada funciona porque A no sabe usar la ruta por defecto (C) sino que hace una petición ARP con una dirección MAC destino desconocida.

Puedes pensar en usar proxy ARP, sin razón porque las respuestas del proxy arp se dan sólo cuando el destino está en un interfaz diferente del origen: ¡Y este no es el caso!

Así que tienes que usar una máscara MUY pequeña (Win0x te permite usar 255.255.255.254, WinNT al menos 255.255.255.248), y tienes que asegurarte de que A y C no tienen la misma dirección de red.

Ejemplos:

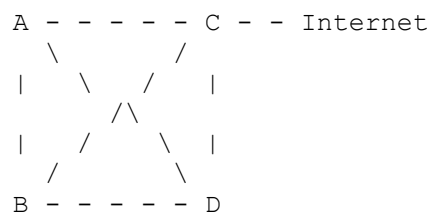
1. IP(A) = x.y.z.2/31, IP(B) = x.y.z.3/31. Esto no funciona porque A pregunta por B en su propia red (petición ARP) y C no responde porque A y B pertenecen al mismo interfaz (así que no va el proxy arp).
2. IP(A) = x.y.z.1/31, IP(B) = x.y.z.2/31. Esto sí funciona porque A pregunta a C (envía peticiones a B con la MAC de C) por B.

En general con una máscara de 255.255.255.254 el sistema funciona con 2 IP cambiando únicamente el bit final.

Esto es TCP/IP forzado pero es la única forma de obtener un alto nivel de flexibilidad.

Nota: Si usas un Punto de Acceso (red en modo infraestructura) no tienes el problema de redirección porque todo se soluciona en el nivel de enlace (casi todos los Puntos de Acceso funcionan como un bridge...). Pero los Puntos de Acceso son caros (sobre 1000 dólares americanos o más) y es más barato usar un P133 con 32 RAM para reenviar incluso con 2 o más tarjetas.

Acceso a Internet

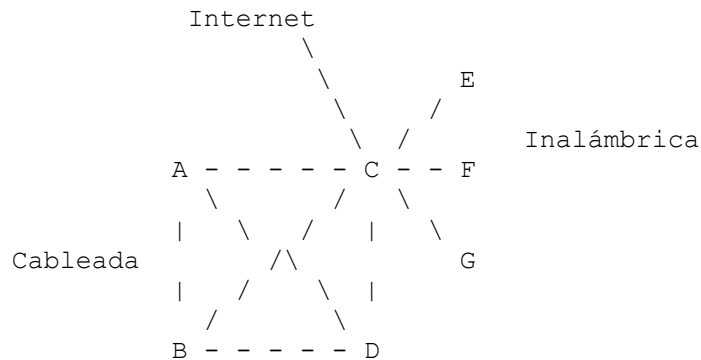


Hay un número de situaciones:

1. C es la única IP pública. Sólo tienes que configurar direcciones privadas (192.168.x.y por ejemplo) para la red inalámbrica habilitando en C el reencaminamiento y el enmascaramiento. A, B y D apuntarán a C.
2. Tienes una máscara pública visible desde Internet y C es tu GW por defecto a Internet para la red. Aquí sólo tienes que habilitar el reenvío en C, configurando C como ruta por defecto para A, B y D.
3. Tienes una máscara visible desde Internet y C no es la pasarela por defecto a Internet. Tienes dos posibles soluciones: Modificar tu ruta por defecto para hacer que sea C. Al mismo tiempo hacer que C apunte a la pasarela a Internet. Puedes,

también, habilitar el proxy arp en C (`echo 1 > /proc/sys/net/ipv4/conf/ethx/proxy_arp` donde ethx es el interfaz hacia la pasarela predeterminada) y configurar tu pasarela en C para que apunte a la pasarela predeterminada. El proxy arp es forzar TCP/IP pero funciona bien.

Mezcla de redes: Cableada e Inalámbrica



Ahora C une dos redes: a la derecha la inalámbrica y la cableada a la izquierda.

Además tienes acceso a Internet por lo que en total tienes 3 tarjetas de red en C.

¿Qué direcciones IP asigno a las máquinas? Tienes dos posibles soluciones:

1. Separa la red en 2 subredes: por ejemplo 192.168.1.0/24 y 192.168.2.0/24. Esta solución es rápida pero no escalable si estás usando direcciones IP de internet porque estarás desperdiciando demasiadas IPs.
2. Habilita el proxy ARP en C para las dos interfaces. Los parámetros de red (dirección de red y máscara) son los mismos para la red inalámbrica y cableada, pero con el proxy-arp habilitado puedo elegir que IPs están en la cableada y cuales en la inalámbrica.

Ahora examinamos la solución 2

Por ejemplo: Considera que tienes una subred pública de internet x.y.z.0/24.

Los interfaces son:

1. `ifconfig eth0 x.y.z.C netmask 255.255.255.255 (cableada)`
2. `ifconfig eth1 x.y.z.C netmask 255.255.255.255 (inalámbrica)`
3. `ifconfig eth2 x.y.z.C netmask 255.255.255.255 (a Internet)`

Rutas estáticas en eth2:

1. `route add IPGW dev eth2`
2. `route add default gw IPGW`

Esta ruta está para direccionar todas las peticiones de Internet a tu pasarela por defecto: como habrás notado, primero le tienes que decir al LINUX donde está el router, entonces hacer las peticiones a través de él.

Rutas estáticas en eth0:

1. `route add x.y.z.A dev eth0`
2. `route add x.y.z.B dev eth0`
3. `route add x.y.z.D dev eth0`

Máquinas A,B y D en la red cableada

Rutas estáticas en eth1:

1. `route add x.y.z.E dev eth1`
2. `route add x.y.z.F dev eth1`
3. `route add x.y.z.G dev eth1`

Máquinas E,F y G en la red inalámbrica

Nota que la flexibilidad es muy alta, pero tienes que configurar manualmente cada máquina.

6. Configuración

Aquí expongo algunos ejemplos (¡espero que útiles!) acerca de como configurar de una forma más difusa y menos en profundidad las tarjetas inalámbricas.

6.1 Información general de configuración

Las tarjetas inalámbricas tienen una interfaz similar a cualquier tarjeta Ethernet, así que tienes que añadir en `/etc/conf.modules`:

1. "alias ethx module", donde ethx es el interfaz que quieres que le sea asignado a tu tarjeta inalámbrica y module es el nombre del módulo del kernel.
2. "options module io=0xAAA irq=I ...", donde 0xAAA es la dirección base IO para asignar a la tarjeta, I es la IRQ e igualmente si hay otros parámetros.

Después de esto podrás usar los comandos `ifconfig` y `route` para configurar tu tarjeta a nivel IP.

6.2 Proxim Symphony

Tipo de red: FHSS, Adhoc sólomente con el protocolo propietario OpenAir.

Web: <http://www.proxim.com/> donde necesitas bajar la documentación y el driver para LINUX y Win9x.

Los drivers vienen con el código fuente a compilar:

1. Deshaz el tar en un directorio vacío
2. teclea `make` para ver la ayuda.

3. `make modules`; `make modules_install` para instalar el driver `rlmod.o` y la utilidad `rl2cfg`.
4. para ejecutar el driver (después de haber modificado el `/etc/conf.modules`: mira Par.6.1.) sólo necesitarás levantar el interfaz con el comando `ifconfig`.

La utilidad `rl2cfg` (para la ayuda teclea `man rl2cfg` una vez que hayas hecho el paso c) te permite cambiar la configuración típica del nivel de enlace:

1. `"rl2cfg dev ethx sta"` para configurar a estación (esclavo)
2. `"rl2cfg dev ethx msta"` para configurar como primaria (Master)
3. `"rl2cfg dev ethx alt"` para configurar en modo automático.

Esto es todo lo que tienes que saber para hacerlo funcionar correctamente.

6.3 Webgear Aviator 2.4 y AviatorPro

Tipo de red: FHSS, Adhoc sólo para Aviator 2.4 e infraestructura para AviatorPro.

Estas tarjetas necesitan más para funcionar, porque tienes que configurar las fuentes PCMCIA porque vienen con conector PCMCIA.

Web <http://www.webgear.com/>.

Para configurar:

1. Tienes que descargarte las fuentes PCMCIA y descomprimirlas en `/usr/src/pcmcia` (ver Par 5.2)
2. Descarga el driver de <http://www.webgear.com/> y teclea `"tar zxvf driver.tgz"` en el directorio `/usr/src/pcmcia`
3. reconfigurar PCMCIA (ver Par 5.2)
4. Siguiendo las instrucciones tendrás que añadir al fichero `/etc/pcmcia/config.opts` una entrada `"source ./ray_cs.opts"`.
5. Date cuenta que en `/etc/pcmcia/ray_cs.opts` hay una línea como ésta: `"module "ray_cs" opts "..."`. Aquí tienes que modificar algunas configuraciones del nivel de enlace presentes en `"opts"`.

Argumentos:

- `pc_debug=x`, donde `x` es el nivel de registro.
- `net_type=x`, `x=0` para AdHoc, `x=1` para Infraestructura.
- `ssid=x`, `x` es el ESSID

Finalmente, para verificar la configuración con el `pc_debug > 0`, verás los mensajes de la capa de enlace en tu consola como estos: `"network started"` para una nueva red inalámbrica creada y `"network joined"` para una nueva red inalámbrica unida a otra.

También el fichero `/proc/ray_cs` puede ayudarte: el BSSID te dice a qué subred pertenece, si es nulo no podrás recibir datos de ninguna tarjeta.

6.4 Productos Lucent Wavelan I, II, Orinoco y Cabletron

Tipo de red: DSSS, Adhoc e infraestructura.

Los productos de Lucent son muy profesionales.

Web: <http://www.lucent.com/> y <http://www.orinoco.net/>.

La configuración es como en WebGear: los pasos 1,2,3 son similares

Luego tienes que añadir al fichero `/etc/pcmcia/config.opts`: `module "wavelan_cs" opts "..."` para Lucent Wavelan I `module "wavelan2_cs" opts "..."`, para Lucent Wavelan II u Orinoco.

En `opts` deberás especificar:

1. `port_type=x`, donde `x` indica Adhoc(3) o Infraestructura(1)
2. `channel=x`, `x=channel`, opción sólo relevante en modo AdHoc.
3. `transmit_rate=x`, para fijar la velocidad: atención a esta configuración por la compatibilidad con las tarjetas Cabletron

Nota: Idealmente es posible en un LiNux tener 2 tarjetas Lucent Wavelanx, una en modo Adhoc y la otra en modo Infraestructura. Sólo una de ellas funcionará adecuadamente porque al arrancar el servicio PCMCIA, las dos tarjetas usarán el mismo valor de `opts` (por lo tanto en modo Adhoc o Infraestructura). Así que tenemos que crear un módulo LiNux (o puede que un programa del modo usuario) que pueda cambiar los parámetros de la capa de enlace en tiempo de ejecución como el modo de acceso y el canal usado en modo Adhoc!

El parámetro del canal se usa generalmente para evitar las interferencias con otros equipos inalámbricos.

Los drivers de Lucent también se pueden usar con tarjetas Cabletron <http://www.cabletron.com/>

6.5 YDI

Tipo de red: DSSS, Adhoc e Infraestructura.

YDI vende tarjetas muy profesionales con antenas, amplificadores y más.

Web <http://www.ydi.com/>

Para instalar:

1. Descomprímelo en un directorio vacío.
2. teclea `make` para compilar.
3. `make install` para instalar el driver `am930_isa` y la utilidad `wlanctl`

Una vez hecho esto puedes elegir si tecleas comandos manuales usando la utilidad de la capa de enlace "wlanctl" o ficheros "scripts/rc.lan" para configurar automáticamente la red.

En el caso manual hay unas configuraciones principales:

1. "wlanctl scan ..." para buscar BSSs presentes.
2. "wlanctl netlist" muestra qué ha encontrado con "wlanctl scan ...".
3. "wlanctl bsscreate ... ssid" para crear una nueva red con ssid parameter.
4. "wlanctl bssjoin bssid" para unirse a la red identificada por bssid.
5. "wlanctl authen" y "wlanctl assoc" para servicios de autenticación.

7. Más sobre la tecnología inalámbrica

7.1 Una distribución LiNIX inalámbrica

La distribución inalámbrica FlyingLinux comenzó en octubre de 1999 en el "Telecommunication Systems Lab" de "Teleinformatics KTH" con el objetivo de estudiar las posibilidades de usar IPv4 Móvil y el acceso inalámbrico estándar basado en DHCP en los laboratorios de los estudiantes.

El resultado de ese trabajo fue que el entorno FlyingLinux estuviera disponible para cientos de estudiantes y profesores durante el curso del proyecto 2G1303 que tuvo lugar de marzo a mayo del año 2000.

FlyingLinux es la primera distribución de LiNIX orientada a los servicios móviles. Hemos tenido cuidado de los aspectos de seguridad incluyendo soporte Kerberos y OpenSSH.

FlyingLinux es parte del movimiento "Open Source". Hemos incluido programas que han sido desarrollados en KTH bajo la licencia GPL.

Puedes encontrar la distribución inalámbrica de LiNIX en este [Web](#).

Desarrollador: Alberto Escudero [Correo-e](#) >, [Página principal](#)

8. Enlaces útiles

8.1 Enlaces de software libre

- [Linux-wlan project](#)
- [Jean Tourrilhes Wireless Howto](#)

8.2 Enlaces comerciales

- [Fatamorgana Computers](#)
- [Web de Lucent](#)
- [Web de YDI](#)

- [Web de Siemens](#)

9. PUF - Preguntas de uso Frecuente - FAQ -

Q1: ¿Cual es la diferencia entre el BSSID y el ESSID y cuando necesito un ESSID?

A1: BSSID es un número de 48 bits usado para identificar la pequeña zona de una BSS, donde todas las máquinas pueden hablar entre ellas (eventualmente con un Punto de Acceso) ESSID es una cadena de longitud variable que permite la comunicación entre diferentes BSS para extender el Conjunto de Servicio Extendido (ESS). Hay un Punto de Acceso por cada BSS y todos ellos hablan sólo si pertenecen al mismo ESSID. Realmente necesitas un ESSID si tienes una red grande con al menos 2 Puntos de Acceso.

Q2: ¿Qué Punto de acceso debo comprar?

A2: El menos caro que puedas encontrar: lo que es importante es que el Punto de acceso y las tarjetas que estás usando usen la misma especificación en la capa física: todas compatibles FHSS o todas compatibles DSSS. Cuidado con las tarjetas Proxim RageLan2 que no pueden hablar con otros FHSS estándar porque usan el protocolo propietario OpenAir.

Q3: ¿Para qué uso la configuración del canal?

A3: Cuando tienes más de una red con diferentes BSS (y diferentes marcas) puedes tener un problema de interferencias: cambiar el canal de los Puntos de Acceso o en el los nodos en modo Adhoc puede ayudarte con este tipo de problemas.

Q4: ¿Por qué no puedo configurar el canal en máquinas con modo Intraestructura?

A4: Porque el canal lo decide el Punto de Acceso.

10. Apéndice A - Máscara 255.255.255.255, proxy arp y bridging

Aquí veremos algunas ventajas en las redes inalámbricas.

LiNIX te permite usar una máscara como 255.255.255.255 para un interfaz lo que puede ayudarte al asignar direcciones IP a cualquier interfaz que quieras, por ejemplo uno en eth0, otro en eth1 y así...

Esto no tiene efectos secundarios especiales.

Además tienes la configuración del proxy arp bajo /proc/sys/net/ipv4/conf/ethx/proxy_arp donde ethx es tu interfaz.

Haciendo "echo 1 > proxy_arp" habilitas el proxy_arp para ese interfaz mientras que con "echo 0 > proxy_arp" lo deshabilitas.

¿Que es proxy_arp? De forma rápida proxy arp te ayuda cuando quieres una respuesta de enrutamiento a una petición ARP si la dirección de destino está en otra interfaz del router LINUX.

Ejemplo:

```
192.168.1.1 ---- 192.168.1.2 router LINUX 192.168.2.2 ----192.168.2.1
```

Para hacer funcionar este ejemplo deberías:

Sin proxy-arp

1. En la máquina 192.168.1.1 poner 192.168.1.2 como enlace por defecto
2. En la máquina 192.168.2.1 poner 192.168.2.2 como puerta de enlace.
3. haciendo pings con éxito desde ambos extremos.

Con proxy-arp

1. En la máquina 192.168.1.1 poner 192.168.1.2 como puerta de enlace
2. no pongas puerta de enlace para la máquina 192.168.2.1 pero habilita el proxy_arp para el interfaz derecho del router.
3. haciendo ping con éxito desde cualquier extremo

Proxy-arp en 2. caso permite al router LINUX contestar cuando tu haces ping desde 192.168.2.1, diciendo que él tiene la máquina 192.168.1.1 por lo que contesta por ella. Después, cuando el origen comienza a enviar paquetes ICMP, el router LINUX sabe que tiene que redirigirlos al destino real 192.168.1.1

En una red inalámbrica el proxy arp puede ayudarte si tienes muchas máquinas LINUX que actúan como repetidores IP y no quieres tener que configurar en todas las máquinas unas rutas estáticas.

También puedes experimentar con el bridging con LINUX en una red inalámbrica:

1. instala un kernel estable reciente
2. descarga los útiles apropiados de bridge de [enlace http](#) o [enlace ftp](#)

El bridging debería ser más sencillo de administrar.

11. Apéndice B - Radio Módem Siemens DECT

Web: <http://www.siemens.com/>

¿Qué es eso? Esos 2 componentes no son realmente tarjetas PC sino que son más como los módems externos.

```
Máquina1-serie - RadioModem1 - - - - - RadioModem2 - serie-Máquina2
```

¿Cómo puedo conectarlo?

Si los ves con una visión abstracta puedes modelarlos como ésto:

```
Máquina1-serie - -CABLE NULL MODEM - - serie-Máquina2
```

Así tienes conexión entre dos puertos serie con dos posibles configuraciones:

1. LINUX con Windows, LINUX tiene un demonio que contesta a una llamada ppp mientras que Windows hace una llamada bajo el acceso remoto.
2. LINUX con LINUX, donde puedes ejecutar (en ambas máquinas) una conexión ppp con direcciones IP inversas.

Para 1 puedes usar este sencillo script en LINUX:

```
"/usr/sbin/pppd -detach lock idle 300 crtscts connect "/usr/sbin/chat -v TIMEOUT 5 AT OK AT OK AT OK AT OK" IPLINUX:IPWINDOWS /dev/ttySx 115200 disconnect "/usr/sbin/chat -v AT OK" ms-dns IPDNS"
```

donde:

- /dev/ttySx es tu puerto serie,
- IPDNS es la dirección IP de tu servidor DNS,
- IPLINUX es la dirección IP de tu LINUX e IPWINDOWS es la dirección IP del Windows.

¡El script anterior se necesita para permitir a Windows creer que hay un modem en el extremo serie!

En Windows necesitas crear una conexión con el crtscts habilitado, velocidad a 115200 con un número cualquiera a llamar (lo necesita el Acceso Remoto pero no se usa en absoluto).

Con 2 máquinas LINUX sólo tienes que ejecutar un script realmente sencillo como éste en cada LINUX:

```
"/usr/sbin/pppd passive local crtscts IPLINUX1:IPLINUX2 /dev/ttySx 115200 noauth persist"
```

donde tendrás que invertir IPLINUX1 con IPLINUX2 en el otro extremo.

Date cuenta de que puedes hacer autenticación tanto en Linux-Windows como en Linux-Linux si quieres.