

Implementación de un hostap como gateway wireless
Iván Belmonte <ttyp0@inet2u.com - <http://asnl.ath.cx>>
v1.1, 18 de Marzo de 2003

Este documento explica el proceso paso a paso que un usuario puede (o no) seguir para implementar una red wireless con un punto de acceso a Internet, y su correspondiente enrutado de direcciones, construido sobre un sistema Linux.

El documento actual es una modificación del primero, a petición de algunos usuarios de la lista de correo <mataro-wireless@lists.eslack.org>, con el fin de corregir ciertos fallos hallados en la primera versión. Evidentemente, continuo estando interesado en recibir las críticas y/o aportaciones de los usuarios que lo lean.

1. Introducción

Una red básicamente se remite al concepto de compartir recursos entre un grupo de dos o más ordenadores. Este concepto no ha cambiado en ningún momento sea cual sea el medio por el que viaje la información a intercambiar, y del mismo modo tampoco ha cambiado el concepto de *gateway* o *router*, que es la máquina (programada manualmente en nuestro caso, o una *caja negra* comprada a un proveedor de material) que nos une a otra red, o en cualquier caso, a INTERNET (nuestro objetivo final).

En nuestro caso, queremos unir uno o más ordenadores a un punto de acceso que nos va a dar conexión a internet, y de momento contamos con dos tarjetas de red inalámbricas, que irán una en el ordenador *cliente* (el nuestro) y otra en el ordenador *enrutador* (el que nos da el acceso a la red, conocido en terminología wireless como **Access Point**). Cabe destacar que la presencia de un Access Point es independiente de la conexión a internet, cosa que veremos en la primera sección de este documento, nada más entrar en materia.

La tecnología wireless está produciendo un gran impacto entre los adictos a la informática por el hecho de posibilitar la movilidad de las estaciones, y consigo, de la información. Por muchos es sabido que también provee al usuario de anonimidad a la hora de realizar tareas indebidas sobre redes ajenas, y quiero aclarar que en este documento no se va a explicar nada acerca del *hacking* de redes inalámbricas, así como tampoco se va a dar detalle alguno acerca de como introducirse en lugares no permitidos. Si es lo que estás buscando, estás en el documento equivocado.

Para entender este documento no es necesario un conocimiento exhaustivo del medio, ni de transmisión de datos, pero sí recomendable tener ciertos conocimientos previos sobre elementos de una red, así como una mínima soltura con sistemas Linux. Si no los poseéis, os recomiendo leer el *Net-3 HOWTO* y perderle el miedo a Linux, que no es tan difícil como lo pintan.

2. Material necesario

Para implementar una red wireless todo el material que necesitamos son dos tarjetas de red wireless, en mi caso particular dos US ROBOTICS USB2415. Con ellas podremos montar la red entre dos portátiles con ranuras PCMCIA... si vais a usar ordenadores de sobremesa obviamente necesitareis su correspondiente adaptador PCI-PCMCIA.

Supondremos que en el ordenador que hará de Access Point hay un módem conectado que da el acceso a Internet, por poner un ejemplo usaremos un módem normal de 56kbps, aunque el caso puede variar según la conexión que cada uno posea... a fin de cuentas no habrá mucha diferencia en la sintaxis a implementar.

Con respecto al software necesario, partiré de la base de que trabajamos con sistemas Linux en ambos bandos, tanto en la estación cliente como en el Access Point. Para implementaciones sobre software propietario, podeis dirigiros a las vías de contacto oficiales del distribuidor de vuestro software.

Las conocidas antenas wireless no son NECESARIAS, aunque pueden aumentar la potencia de emisión de vuestras tarjetas. En el documento voy a prescindir de ellas, el único cambio de enchufar una antena a no enchufarla será el radio de movilidad de las estaciones, y su aplicación no comporta modificación alguna a la implementación del software, de modo que esta opción queda abierta a gusto de cada uno.

Advertencia: Es MUY importante que os asegureis de que la tarjeta que pretendéis usar en el lado del Access Point lleva chipset **PRISM-2 / PRISM-2.5 / PRISM-3**. De lo contrario no podreis usarla como Access Point.

3. Primer paso, configuración del núcleo

Hemos llegado a casa después de un duro día de trabajo, y rápidamente desenfundamos nuestras nuevas y flamantes tarjetas. Abrimos la caja de una de ellas y nos dirigimos a la estación cliente (un portátil en mi caso). Con las prisas sacamos la tarjeta de la caja y dejamos a un lado los cartones, bolsitas Y MANUALES de la tarjeta. Introducimos la susodicha en la ranura PCMCIA y... tachan!!! ... ¡¿queda chula eh?! ... vale, ahora ya podemos volver a darnos la vuelta y a buscar los manuales de la tarjeta. Sería bueno echarles una ojeada...

NOTA: la configuración del núcleo ES NECESARIA tanto en las estaciones cliente como en la estación que haga de punto de acceso

Partimos de un sistema Linux con una consola (bash por ejemplo), el entorno gráfico no afecta al rendimiento de la tarjeta (lo juro), por lo que cada uno que trabaje sobre el que le parezca más bonito. Nos situamos en el directorio raíz de los fuentes del núcleo, y vamos a echar una ojeada a la configuración del mismo para adaptarlo a nuestras necesidades, qué emocionante:

```
# cd /usr/src/linux
# make menuconfig
```

Dándole vueltas encontraremos soporte para todo tipo de hardware, y yo no tengo tiempo material ni conocimiento exhaustivo de todos los componentes que existen en el mercado, de modo que no voy a especificaros todas las opciones que debéis marcar y/o desmarcar. Me limitaré a especificar lo estrictamente necesario para nuestra árdua tarea:

1) Vamos a General Setup

- Dentro, buscamos la opción PCMCIA SUPPORT, y entramos dentro
- Deshabilitamos el soporte PCMCIA

2) Vamos a Network Device Support

- Dentro, buscamos la opción WIRELESS LAN (NON-HAMRADIO)
- Habilitamos la opción WIRELESS LAN (NON-HAMRADIO)
- No habilitamos ningún módulo dentro de esta opción, y si hay alguno habilitado, lo deshabilitamos

3) **(Sólo necesario en la estación que haga de Punto de Acceso)** Antes de seguir, y en vistas a lo que necesitaremos más adelante (capítulo 5) para realizar el enrutamiento, os recomiendo dar soporte a todos los módulos dentro de la sección NETWORKING OPTIONS --> NETFILTER CONFIGURATION

De este modo vuestro sistema ya estará preparado para ejercer las funciones de enrutador (NAT - Network Address Translation).

4) Vamos a SAVE CONFIGURATION TO ALTERNATE FILE, y con esa opción podremos guardar la nueva configuración en un fichero con un nombre que nos resulte amigable, para posibles pérdidas futuras tras las que queramos recuperar la configuración correcta.

5) Salimos del menu y respondemos afirmativamente cuando nos pregunte si queremos guardar la configuración del núcleo.

Bien, ya tenemos una configuración del núcleo que nos permitirá establecer una red wireless, pero si os fijáis hemos deshabilitado el soporte para tarjetas PCMCIA. ¡¡Anda!! pero si vuestras tarjetas son PCMCIA... sí, es cierto, pero luego compilaremos aparte los módulos necesarios para utilizarlas. Ahora solo falta compilar el nuevo núcleo y sus módulos:

```
# make dep && make clean && make bzImage  
# make modules && make modules_install
```

Si en algún punto la compilación falla, fijaos en los paths del gcc para ver lo que estaba compilando, y volved al principio de la configuración del kernel. Posiblemente os falten o sobren opciones y por eso falle. Si deseais adquirir un conocimiento mínimamente exhaustivo sobre el núcleo (kernel) y su manipulación, podeis dirigiros al KERNEL HOW-TO (<http://www.tldp.org> , ahí encontrareis todos los HOWTO's existentes).

```
# cp /usr/src/linux/arch/i386/boot/bzImage /boot/bzImage  
# cp /usr/src/linux/System.map /boot/System.map
```

Ahora debéis retocar vuestro gestor de arranque para que os de la posibilidad de arrancar usando la nueva imagen:

- Usuarios de GRUB: vim /boot/grub/grub.conf
- Usuarios de LILO: vim /etc/lilo.conf; lilo

Perfecto! ya teneis soporte para redes wireless en vuestras estaciones... pero, ¿y qué pasa con las tarjetas?

4. Comenzando (Client point side...)

Bien, el núcleo de Linux no tiene su fuerte precisamente en el soporte para tarjetas PCMCIA porque hay muchas tarjetas que no están soportadas, así que usaremos un paquete externo que compila módulos de tarjetas PCMCIA y los incluye al núcleo. En ese paquete SI estan soportadas nuestras tarjetas, y para evitar disfunciones entre módulos lo que hacemos es usar únicamente el soporte compilado por este paquete externo, quitando el soporte del núcleo.

El paquete se llama *pcmcia-cs*, y podeis encontrarlo en el CD de instalación de vuestra distribución, o descargarlo directamente de la página web:

<http://pcmcia-cs.sourceforge.net/>

Los usuarios de Gentoo Linux simplemente debeis ejecutar el correspondiente *emerge* (*emerge pcmcia-cs*). Los usuarios de otras distribuciones os bajareis el paquete y lo compilareis a mano si no lo quereis instalar directamente del CD:

```
# cd /usr/local/src
# wget -c http://easynews.dl.sourceforge.net/sourceforge/pcmcia-cs/pcmcia-cs-3.2.4.tar.gz
# tar zxvfp pcmcia-cs-3.2.4.tar.gz
# cd pcmcia-cs-3.2.4
# make config
# make all
# make install
```

Eso compilará los módulos para todas las tarjetas PCMCIA actualmente soportadas (la grandiosa mayoría), y los copiará en el árbol de directorios de módulos del kernel, bajo el directorio */lib/modules/`uname -r`/pcmcia* (*uname -r* es la versión del kernel).

Ahora SI que teneis soporte para tarjetas PCMCIA, y vuestro núcleo tiene soporte para redes wireless, de modo que ya podemos proceder al temido comando que nos mostrará si todo ha ido bien:

```
# shutdown -r now
```

o simplemente...

```
# reboot
```

Si todo ha ido bien, ahora vuestro sistema debería arrancar con el kernel nuevo, y tras logear en el sistema podreis comprobarlo de la siguiente forma:

```
# modprobe pcmcia_core
# modprobe i82365
# modprobe ds
# cardmgr -f
# ifconfig eth0 192.168.2.1
# ifconfig -a
```

Ahora vuestra tarjeta debería activarse, y la salida del último comando os debería informar de que efectivamente la interface eth0 eá UP and RUNNING. Si algo ha fallado empezad desde la configuración del kernel y revisadla bien. En caso de que os haya salido todo bien, podeis soltar el teclado y saltar un rato por la habitación: ya teneis un ordenador capaz de conectarse a una red wireless.

5. Continuando... (Access Point side)

Muy bien, ahora ya podeis conectaros a un nodo wireless... pero, ¿a qué nodo? Ahora cogemos la silla y la arrastramos hasta el ordenador que ha de hacer de Access Point. Los pasos que vamos a realizar son exactamente los mismos que en el caso anterior, con una pequeña variación, y es que tenemos que dar soporte a la tarjeta para ejercer de Access Point, lo que se realiza con un módulo externo (como los de las PCMCIA's, se compila aparte del núcleo) llamado *hostap*. Podeis descargar el módulo *hostap* ANTES de compilar *pcmcia-cs*, para incluir el fuente del módulo junto con el resto de módulos de *pcmcia-cs*, y compilarlo todo del tirón, o podeis hacerlo por separado.

Los usuarios de Gentoo Linux lo tienen fácil: *emerge hostap*, los de otras distribuciones deberán compilarlo a mano. Podeis encontrarlo en <http://hostap.epitest.fi>:

```
# cd /usr/local/src
# wget -c http://hostap.epitest.fi/releases/hostap-2002-10-12.tar.gz
# tar zxvfp hostap-2002-10-12.tar.gz
# cd hostap-2002-10-12
# cp driver/* /usr/local/src/pcmcia-cs-3.2.4/
# cd /usr/local/src/pcmcia-cs-3.2.4
# make config
# make all
# make install
```

Si preferis compilar el modulo *hostap* por separado (por ejemplo porque el paquete *pcmcia-cs* ya lo habiais instalado de serie con la distribución (no trae *hostap* incorporado)), pues simplemente:

```
# cd /usr/local/src
# wget -c http://hostap.epitest.fi/releases/hostap-2002-10-12.tar.gz
# tar zxvfp hostap-2002-10-12.tar.gz
# cd hostap-2002-10-12
# make hostap_plx
# make install_plx
```

IMPORTANTE: *EL módulo a cargar depende del tipo de tarjeta que poseais. Existe la posibilidad de que tengais 3 tipos diferentes de conexión para la tarjeta:*

*1) Slot pccard ,las típicas ranuras en el portatil. En este caso vuestro modulo se llama **hostap_cs**.*

*2) Adaptador PCI-PCMCIA (PLX9050). Es una tarjeta PCI para pinchar a un ordenador de sobremesa, que lleva una ranura en la que enchufar la PCMCIA. En este caso el módulo a usar es el que el ejemplo expone (por ser mi caso particular): **hostap_plx**.*

3) Tarjeta PCI normal y corriente (basadas en chip PRISM-2.5), sin adaptador ni nada que precise de la presencia de una PCMCIA. En este caso usareis *hostap_pci*.

De este modo, seguid el ejemplo sustituyendo únicamente el nombre del módulo a compilar e instalar por el que vuestro caso requiera.

En este punto debo advertiros que yo, al montar mi hostap en Slackware me encontré con problemas por compilar el pcmcia-cs con hostap dentro, de modo que lo hice por separado y me funcionó perfectamente bien.

Bien, tras haber compilado primero el kernel y luego realizado estos últimos pasos, rebotamos la máquina como en el caso anterior, Una vez rebotada y logeados en el sistema:

```
# modprobe pcmcia_core
# modprobe i82365
# modprobe ds
# cardmgr -f
```

Teóricamente debería funcionaros a la perfección... yo tuve problemas y no me funcionó, ya que me daba errores en la carga de los módulos para PCMCIA. Mi solución fue la siguiente:

```
# modprobe pcmcia_core
# modprobe hostap_plx
```

Y A RULAR!

Sea como sea que lo hayais hecho, si os ha funcionado bien, deberiais poder ver la interface de red wireless a la salida del ifconfig, en este caso a mí me figura como *wlan0*:

```
# ifconfig wlan0 192.168.2.254
# ifconfig -a
```

Como en el caso anterior, deberiais ver la interface UP and RUNNING. Ahora ya podeis probar de tirar pings (por ejemplo) de una máquina a la otra, y sin ningún problema tienen que poderse ver... ¡MUY BIEN! ya teneis un punto de acceso y un cliente... ahora ya podeis llamar a los amigos y montar una wireless party contra vuestro punto de acceso, que de momento no os lleva a Internet pero os permite veros sin tener que tirar metros de cable ni turnaros las bocas del hub un ratito cada uno.

Ya tenemos completado el segundo paso... ánimo que ya está casi todo. De momento podeis jugar con vuestra nueva red, que son las 5:00 de la mañana y yo llevo muchos días sin dormir, así que de momento aquí os dejo hasta mañana.

6. Enrutamiento

Seguramente no os conformareis con mantener una red wireless tal cual, sino que además querreis poder acceder a Internet desde las máquinas que se conecten al Access Point. No es un paso complicado, seguid leyendo...

6.1 Preliminar

Posiblemente tengais en casa una conexión con ADSL o cable de fibra óptica. Otros teneis un módem de 56kbps. La verdad es que a efectos de configuración de nuestra red eso importa poco, tan solo afectará al nombre de la interface a enrutar.

Si teneis un modem de 56kbps, vuestra interface se llamará seguramente **ppp0**, mientras que si teneis una conexión con ADSL o Cable de fibra óptica, seguramente os unireis al cablemodem o al router mediante una tarjeta de red normal y corriente y un RJ45, en ese caso vuestra interface va a llamarse seguramente **eth0**.

Podeis comprobarlo con la salida del siguiente comando:

```
# infconfig -a
```

Yo voy a suponer que os conectais a internet con unmodem de 56kbps, y por tanto usaré como nombre de interface externa **ppp0**. Si alguien se hace un lío con todas estas cosas, le recomiendo echar una ojeada al *Net-3 HOWTO*.

6.2 ¿Qué es enrutar y para que quiero enrutarme?

El término enrutar hace referencia a enviar la información por una ruta específica. Obviamente en las redes suelen haber más de 2 ordenadores, por lo que parecería lógico enchufar un módem a cada ordenador para darle una conexión a Internet. Lógico, pero no coherente.

En el momento en que damos la conexión a Internet a un ordenador, este recibe una *IP pública*, esto es, visible desde internet. El resto de ordenadores en nuestra red utilizan *IP's privadas* lo que supone que son inexistentes o *prohibidas* en Internet, de modo que tendrán que usar la única *IP pública* dela red para salir a internet.

La trayectoria de la información es sencilla: nuestras estaciones de trabajo envían la información al ordenador que posee la conexión a internet (denominado *gateway*), que está dotado de una ip privada para comunicarse con las máquinas internas y de una ip pública para comunicarse con el resto del mundo. Éste, una vez recibe la información analiza las cabeceras de los paquetes que ha recibido, y observa el destino de los mismos. Como el destino no se encuentra en la red interna, los enviará a través de la interface con *IP pública* a Internet.

Es necesario entender este concepto, y saber que en las cabeceras de los paquetes de información se encuentra la información referente al origen y destino de los mismos.

Bien, ¿qué pasará entonces una vez el destino haya recibido la información y responda a quien se la ha enviado? Pues era de esperar... la dirección IP de origen era privada, y como he explicado anteriormente, prohibida en internet. De modo que esos paquetes nunca llegarán de vuelta a casa, y la comunicación será imposible.

¿Qué hace un enrutador para permitir la comunicación entre máquinas internas y máquinas externas a nuestra red? Cuando recibe un paquete que tiene que viajar a internet, cambia en su cabecera la dirección IP de origen (privada) por la de su interface externa (pública), de modo que el receptor sepa a quién contestar una vez procesada la petición. A la vuelta de la información, nuestro enrutador observará que le llegan paquetes destinados a sí mismo, pero mirando en su *registro* de enrutamientos, sabrá que él no esperaba recibir nada de nadie, y que esos paquetes son para la máquina que en un principio había iniciado la transmisión, de modo que cambiará la IP de destino (la suya) por la IP de la máquina que espera la respuesta (privada), y lo enviará hacia dentro de la red. Así la transmisión de paquetes entre el origen y el destino habrá sido satisfactoria.

6.3 Enrutar con Linux

Para llevar a cabo el enrutamiento con Linux se necesita un paquete llamado *iptables* que posiblemente os habrá instalado vuestra distribución de serie. Si no es así, podeis descargarlo de internet, se encuentra en la siguiente dirección:

<http://www.iptables.org/>

Si quereis saber si lo teneis ya instalado, podeis ejecutar simplemente:

```
# iptables
```

Si la respuesta es del tipo "*eing?*" ya sabeis que no lo teneis instalado. Si la respuesta es del tipo "*faltan parámetros*" sabeis que sí lo teneis instalado.

Iptables es la herramienta que usa *Netfilter* (www.netfilter.org) para ejercer un control sobre el tráfico de información, de modo que nos permite tanto enrutar como filtrar, las herramientas necesarias para construir un firewall, aunque yo voy a limitarme en este documento a explicar solo lo que a enrutado concierne dado que de lo contrario el texto se haría demasiado extenso.

Partiendo de un kernel con soporte de iptables y todos sus modulos compilados, las líneas que debeis teclear para habilitar el enrutado son las siguientes:

```
# modprobe iptable_nat
# modprobe ip_conntrack_ftp
# modprobe ip_conntrack_irc
# modprobe ip_nat_ftp
# modprobe ip_nat_irc

# iptables -t nat -F PREROUTING
# iptables -t nat -F POSTROUTING
# iptables -t nat -F OUTPUT

# iptables -t nat -A POSTROUTING -o ppp0 -s 192.168.2.0/24 --to-source
217.126.51.143
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Las primeras líneas son para cargar los módulos necesarios para habilitar NAT y algunas facilidades como el enrutamiento de los protocolos IRC y FTP, que sin esos módulos van a tener problemas para ser enrutados. Las siguientes líneas son para vaciar el estado actual del filtrado, por si hubiese algún filtro ejecutándose que pudiera ocasionar que algo no nos funcionase bien. Las últimas dos líneas son las que habilitan la función de enrutamiento en nuestra máquina. La primera de ellas contiene algunas cosas que debereis sustituir por el valor correspondiente en cada caso:

ppp0 -> vuestra interface externa

192.168.2.0/24 -> el rango de vuestra red local (192.168.1.0/24, 172.26.0.0/24, 10.10.0.0/24....)

217.126.51.143 -> vuestra ip externa (la que el módem os haya asignado)

Obviamente escribir todas estas líneas a mano cada vez es un engorro, por lo que os recomiendo que escribais estos comandos secuencialmente en un fichero de texto que podais ejecutar cómodamente cada vez que arranqueis el punto de acceso:

```
# vim enrutador.sh
# chmod +x enrutador.sh
```

Si lo haceis de esta manera, los que usais módem de 56kbps, podeis crear este fichero bajo el nombre *ip-up* dentro del directorio */etc/ppp*, y de este modo se ejecutará cada vez que conecteis a internet. También en este caso podreis sustituir el valor de la dirección ip externa por un simple *\$4*, que os cojerá automáticamente el valor de la IP asignada por vuestro proveedor. De lo contrario, cada vez tendreis que sacarla a mano con un *ifconfig*, o generar un pequeño script que la saque del mismo comando el solito.

7. Lecturas interesantes

Como lecturas que considero de interés común a cualquiera que realice tareas en entorno de redes (del tipo que sea), cabe destacar las siguientes:

Net-3 HOWTO

Linux 2.4 Packet Filtering HOWTO

Linux 2.4 NAT HOWTO

(IN)Seguridad en redes Wireless - Pau Oliva (<http://pof.eslack.org/wireless>)

8. Agradecimientos

Agradezco a la asociación **Mataró Wireless** (<http://www.matarowireless.net>) el apoyo y los medios en mi tarea de implementar una red wireless particular. Agradezco también a la **ASSL** (<http://assl.ath.cx>) el compañerismo, la ambición y las ganas de aprender y crear que me ha infundido.

9. Have Fun

Este texto debe llegar a manos cualquiera que lo precise, no tiene derechos de copia ni distribución algunos, y es de un carácter completamente libre. Ruego se me notifique cualquier error en el mismo, así como las posibles aportaciones para ampliarlo a una segunda versión.

EOT