

Compartir Conexión de Cable-COMO

Ricardo Cervera Navarro

ricardo@zonasiete.org

Este documento describe cómo configurar una conexión de cable en una máquina Linux y cómo compartirla con otros equipos de su red local.

1. Introducción

1.1. Sobre este documento

No es un documento extenso ni con demasiados detalles. No es el único que describe este procedimiento y seguramente no sea el mejor. Léalo bajo su propia responsabilidad ;-)

Para cualquier descubrimiento de un error o mejora, por favor póngase en contacto conmigo, y si no le respondo, sírvase usted mismo de redistribuir una versión mejorada de este documento.

1.2. Agradecimientos

A los programadores que han desarrollado todas las maravillosas utilidades que vamos a utilizar, y a los que las han documentado. A Ferdy por despejar algunas de mis dudas (como siempre). A mis compañeros de ZonaSiete.ORG.

2. Descripción de la situación

Tenemos una máquina con Linux, conectada a una red local con otras máquinas (con las que queremos compartir la conexión) y un cable módem.

En la máquina Linux necesitaremos dos tarjetas de red; una será la que se conectará al hub/switch con el resto de máquinas y la otra al cable módem.

De ahora en adelante supondremos que la que se conecta al cable módem es `eth0` y la que se conecta a la red local `eth1`.

La situación podría ser como la siguiente:

4. Conceptos

Necesitamos entender qué vamos a hacer y qué software es el encargado de cada tarea.

En un principio, la máquina Linux podría tener conexión a Internet, pero nosotros lo que queremos es que toda nuestra red local tenga conexión a Internet. La máquina Linux poseerá la conexión a Internet y la compartirá con el resto de máquinas (deberá estar encendida para poder conectar a Internet con otras máquinas de la red local). Además, realizará otros servicios, por lo que es conveniente encenderla antes que cualquier otro cliente de la red. El software encargado de compartir la conexión será iptables/netfilter. Tendremos que cargar los módulos adecuados del kernel y luego establecer las reglas con el comando **iptables** como ya veremos.

Un cable módem nos ofrece una conexión bastante "directa" con Internet, así que necesitamos seguridad extra, proporcionada por un firewall, que en este caso también es netfilter/iptables.

Asignar IPs fijas a las máquinas de la red es algo incómodo; cada vez que se añada una nueva hay que buscar una dirección IP local que no esté siendo usada por ninguna otra máquina... así que mejor dejaremos que la máquina Linux también asigne las direcciones de nuestra red por nosotros; pudiendo estas cambiar en cada arranque de una misma máquina cliente (direcciones dinámicas). El protocolo se llama DHCP y la implementación que usaremos es DHCPd. Asimismo, la propia máquina Linux también será cliente DHCP del cable módem (en la interfaz de red `eth0`), que le asignará una dirección IP válida en Internet. No obstante, la dirección IP de `eth1` será fija, 192.168.0.1, que será la dirección que le daremos a las otras máquinas para que la usen como puerta de enlace o gateway.

Resumiendo esto anterior tenemos que en la máquina Linux, `eth0` tendrá una IP dinámica que es válida en Internet. `eth1` tendrá la dirección 192.168.0.1 y las máquinas clientes tendrán direcciones dinámicas.

El servicio DNS se encarga de traducir los nombres en IPs y viceversa. Son famosas las "caídas" de los servidores DNS de los ISPs, así que nuestra máquina Linux también hará de servidor DNS para las máquinas de nuestra red local. El software encargado de hacer esto se llama BIND.

Veamos qué pasará cuando una máquina cliente haga una petición a Internet; supongamos que la máquina cliente tiene dirección IP 192.168.0.8 y que el cable módem asignó a `eth0` la dirección 1.2.3.4 (que sería válida en Internet). Cuando una máquina cliente haga una petición a Internet, esta petición será recibida por netfilter/iptables y le cambiará la dirección de origen de 192.168.0.8 a 1.2.3.4 y la enviará al sitio de Internet que corresponda. Asimismo, recordará qué máquina de la red local le hizo la petición y cuando vuelva la respuesta de Internet, se la devolverá a la máquina cliente. Este mecanismo se llama NAT (Network Address Translation) y nos permite que varias máquinas compartan una única conexión a Internet, y es exactamente lo que hacen los routers de red (en este caso nuestra máquina Linux se ha convertido también en un router o enrutador).

5. Configuración del software

En primer lugar, instala los paquetes binarios de tu distribución (RPM o DEB o lo que corresponda) de BIND e iptables. iptables no necesita configuración, y la de BIND se comentará brevemente más adelante. Para instalar DHCPd basta con descomprimir el archivo (**tar -xvzf dhcpd....tar.gz**) y dentro de el directorio hacer **./configure && make && make install**.

Crearemos un script que se ejecute en cada inicio y que nos cargue el o los módulos de las tarjetas de red y que arranque DHCP para la asignación de direcciones en las máquinas locales. También cogerá `eth0` la IP que le asigne el cable módem.

```
#!/bin/bash
#### /etc/init.d/dhcp.sh

# Modulo o modulos de las tarjetas (sustituir por el/los adecuado/s)
modprobe 8139too

# Inicializar tarjeta de red eth1
ifconfig eth1 192.168.0.1

# Línea necesaria para DHCP
route add -host 255.255.255.255 dev eth1

# Inicio de DHCP (servidor)
dhcpd -d -f eth1 > /var/log/dhcpd.log 2>&1 &

# Inicio de DHCP (del cable modem)
dhclient eth0
```

Ahora ya tenemos direcciones en las dos tarjetas. Necesitamos habilitar NAT para que el resto puedan conectar, y poner las reglas de filtrado del cortafuegos. Para esto último pensemos... lo "no confiable" es lo que venga de eth0 que sean conexiones "nuevas", las que sean "respuestas" a nuestras peticiones sí las queremos.

```
#!/bin/bash
#### /etc/init.d/nat.sh

# Interfaces
IF_EXT="eth0"
IF_INT="eth1"

# Puertos
P_WEB="8080"
P_FTP="2121"

# IPs
IP_INT="192.168.0.1"

# Modulos
modprobe iptable_nat ip_conntrack_ftp ip_nat_ftp \
        ip_conntrack_irc ip_nat_irc iptable_filter \
        ipt_MASQUERADE ipt_state ip_tables

# Reglas por defecto
iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD
iptables -t nat -F
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT

# Reglas de router (SNAT)
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o $IF_EXT -j MASQUERADE
```

```
iptables -t nat -A POSTROUTING -s 127.0.0.0/24 -o $IF_EXT -j MASQUERADE

# Reglas de servidores (DNAT)
iptables -A PREROUTING -t nat -p tcp -i $IF_EXT --destination-port $P_WEB \
-j DNAT --to $IP_INT:80
iptables -A PREROUTING -t nat -p tcp -i $IF_EXT --destination-port $P_FTP \
-j DNAT --to $IP_INT:21

# Filtrado de paquetes
iptables -A INPUT -i $IF_EXT -p tcp --destination-port 80 -j ACCEPT
iptables -A INPUT -i $IF_EXT -p tcp --destination-port 21 -j ACCEPT
iptables -A INPUT -i $IF_EXT -m state --state NEW,INVALID -j DROP
iptables -A FORWARD -i $IF_EXT -p tcp --destination-port 80 -j ACCEPT
iptables -A FORWARD -i $IF_EXT -p tcp --destination-port 21 -j ACCEPT
iptables -A FORWARD -i $IF_EXT -m state --state NEW,INVALID -j DROP

# Reenvio de IP
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Las líneas de DNAT(PREROUTING) son por si queremos tener un servidor web y ftp accesible desde Internet escuchando externamente en \$P_WEB y \$P_FTP respectivamente (internamente en 80 y 21). En mi caso he puesto 8080 y 2121 porque los puertos 1024 e inferiores son bloqueados por mi proveedor de Internet. Si no estamos interesados en ello podemos comentarlas, poniéndoles un # delante. Para entender más a fondo este script consulte la documentación de netfilter/iptables (<http://www.netfilter.org>).

Antes de arrancar el script del servidor DHCP, éste necesita una ligera configuración:

```
default-lease-time          600;
max-lease-time              7200;
option subnet-mask          255.255.255.0;
option broadcast-address    192.168.0.255;
option routers              192.168.0.1;
option domain-name-servers  192.168.0.1;
option domain-name          "dominiored";

subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.2 192.168.0.100;
}

ddns-update-style ad-hoc;
```

Si tiene muchos clientes puede aumentar el rango de direcciones. A continuación instalaremos el paquete BIND. Su única misión será la de un servidor DNS de caché, esto es, simplemente traducirá los nombres de máquinas de Internet en IPs. Es muy común que con la configuración con la que se instala BIND en los paquetes de las distribuciones funcione bien para nuestros propósitos. Si no es así, ahí va una configuración que debería funcionar:

```
#### /etc/hosts
127.0.0.1 localhost
192.168.0.1 router.dominiored

#### /etc/hostname
```

```
router

#### /etc/named.conf

options {
directory "/var/named";
};

zone "." {
type hint;
file "named.ca";
};

zone "dominiored" {
type master;
file "named.dominiored";
};

zone "1.0.0.127.in-addr.arpa" {
type master;
file "named.127.0.0.1";
}

zone "0.168.192.in-addr.arpa" {
type master;
file "named.192.168.0";
}

#### /var/named/named.dominiored
$TTL 259200
@ IN SOA router.dominiored. root.router.dominiored. (
2003100201 ; Serial AAAAMDDNN , NN numero de serie
1800 ; Refresco
3600 ; Reintento
3600000 ; Caducidad
259200 ) ; Tiempo de vida mínimo

IN NS router.dominiored.

router IN A 192.168.0.1

#### /var/named/named.127.0.0.1
$TTL 259200
@ IN SOA localhost. root@localhost. (
2003100201
1800
3600
3600000
259200 )
```

```
IN NS localhost.
IN PTR localhost.

#### /var/named/named.192.168.0
$TTL 259200
@ IN SOA router.dominioed. root.router.dominioed. (
2003100201
1800
3600
3600000
259200 )

IN NS router.dominioed.

1 IN PTR router.dominioed.
```

Nos queda un pequeño fichero de configuración, puesto que la máquina Linux es a su vez cliente DHCP del módem:

```
### /etc/dhclient.conf

interface "eth0" {
    send dhcp-client-identifier 1:00:48:54:88:E7:42;
    send dhcp-lease-time 86400;
}
```

El numerito hexadecimal es distinto para cada tarjeta (también llamado *dirección MAC*). Para obtener el de la suya, haga:

```
# ifconfig eth0 | grep HWaddr | cut -d " " -f 11
00:48:54:88:E7:42
```

A esto añádale "1:" delante exactamente igual que está en el ejemplo.

Para hacer que los scripts se ejecuten al inicio, puede usar las herramientas específicas de su distribución, o bien averiguar su runlevel de inicio con **cat /etc/inittab | grep initdefault | cut -d : -f 2**, y tras saberlo:

```
# ln -s /etc/init.d/dhcp.sh /etc/rcNUM.d/S20dhcp
# ln -s /etc/init.d/nat.sh /etc/rcNUM.d/S21nat
```

donde NUM es el número que averiguamos antes. Ya que estamos configurando manualmente las tarjetas de red y el firewall, sería muy aconsejable que desactivásemos del inicio la configuración que hace cada distribución de iptables y de la red. BIND, al instalarlo desde paquetes, debería arrancar solo automáticamente.

Una buena idea es también instalar un servidor proxy para acelerar la navegación web. Describiría cómo hacerlo aquí, pero ciertamente estaría perdiendo el tiempo, porque Ricardo Soria ya lo hizo bastante bien en su Proxy-COMO (<http://www.zonasiete.org/docs>).

Ahora podemos encender el cable módem, esperar un poco y arrancar los scripts `dhcp.sh` y `nat.sh`, y reiniciar BIND (`/etc/init.d/named restart` o quizás en vez de `named` sea `bind` o `bind9`).

5.1. Configuración de los clientes

En los clientes que sean Linux, detendremos la configuración de la red y la configuraremos manualmente. Para ello hay que seguir los mismos pasos que hicimos con el servidor para crear un `dhclient.conf`. Luego crearemos un script que se ejecute al inicio que cargue el módulo de la tarjeta de red y que haga: **`dhclient eth0 &`**.

En los clientes que sean Windows, en las propiedades de la conexión de red activaremos "Obtener una dirección IP automáticamente" e igualmente para los DNSs.

Si instalamos un servidor proxy, hay que recordar actualizar la configuración de los navegadores para que lo usen.

5.2. Gestión del ancho de banda

Las configuraciones que pongo aquí están sacadas y profunda y adecuadamente explicadas en el *Linux Advanced Routing and Traffic Control HOWTO*. Léalo si no tiene nada mejor que hacer, es muy interesante.

En resumidas cuentas, regular la cantidad de información que nos llega de Internet es complejo y queda pendiente de añadir en este documento (si le apetece, añádale, ponga su nombre y redistribuya este documento bajo las condiciones de la GFDL). Lo que sí que podemos hacer es "repartir" el ancho de banda entre todas las conexiones locales activas con el router Linux. ¿Cómo hacemos esto? La solución es crear una "cola" de paquetes por cada conexión local activa con el router, y mandar los paquetes a Internet por turno entre las distintas conexiones. Por la propia naturaleza del protocolo TCP/IP, "repartir" el ancho de banda de subida implica que se reparte también el ancho de banda de bajada (pero en esta configuración NO se limita).

Debe hacer que los siguientes comandos se ejecuten en cada inicio:

```
tc qdisc add dev eth0 root handle 1: prio

tc qdisc add dev eth0 parent 1:1 handle 10: sfq
tc qdisc add dev eth0 parent 1:2 handle 20: tbf rate 128kbit buffer 1600 limit 3000
tc qdisc add dev eth0 parent 1:3 handle 30: sfq
```

Ahora cada conexión que se mantenga con el router enviará un paquete por turno riguroso, así que nadie podrá hacer que incluso una pequeña petición web se demore en su salida. Sustituya 128 por el valor de subida de su conexión menos un 10 o un 12 por ciento.

NOTA: Esta configuración puede no ser adecuada si nuestros clientes usarán aplicaciones P2P o similares que hacen muchísimas conexiones distintas.

5.3. Documentación adicional

DHCP puede hacer algunas otras cosas interesantes como por ejemplo asignar siempre la misma IP a un ordenador (por ejemplo un servidor, que interesa que siempre tenga la misma dirección); igualmente que netfilter/ iptables. La documentación de estos dos paquetes puede ser de mucha utilidad para casos más particulares que el expuesto en este documento.

Con respecto a la limitación y gestión del ancho de banda, aparte del ya mencionado HOWTO, están también el Bandwidth-Limiting HOWTO y el ADSL-Bandwidth-Management HOWTO. Si cree que este

documento se puede mejorar con configuraciones que usted ha comprobado que funcionan, por favor modifíquelo y redistribúyalo o hágamelo saber.

Sobre iptables tanto para firewall como para masquerading/NAT, puede encontrar más detalles en la web de iptables/netfilter (<http://www.netfilter.org/>).