Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Enrutado en base a marcas de paquetes. Iproute + Iptables. (34833 lectures)

Per **Xisco Fernandez**, ***Gravis*** ()

Creado el 28/11/2002 21:29 modificado el 29/11/2002 02:03

En este artículo se intentará detallar el proceso para realizar balanceo de carga, y discriminación de paquetes en función a las marcas que apliquemos sobre ellos.

Hello world! ;)

Hola a todos, este es mi primer artículo, osea que sed buenos conmigo.

Voy a tratar de explicar de la forma más precisa y concisa, los pasos que seguí para configurar una Debian con 4 interfaces de red, para hacer balanceo de carga y redirección en base al puerto de destino.

En realidad esta máquina podria realizar las mismas funciones con 2 interfaces a base de binds, pero lo hice de este modo por si algún día tuviese que hacerlo con más de 2.

Situación:

Local de ocio sin ánimo de lucro, unos 20 usuarios, 3 salidas a internet. Un cable modem (300Kbps), y dos cable routers cisco ubr954, uno de ellos de 512Kbps y el otro un 300Kbps.

En un principio los cable routers estaban en la misma red que la LAN, 172.31.0.0. Los usuarios podían cambiar su puerta de enlace dependiendo de lo que quisieran hacer. A veces un cable router tenía una ruta (asignada por ono) que favorecía a los juegos on line, en este caso quake3. Se que os sonara extraño esto último, pero es cierto, en el caso de los cable modem, ono, en funcion de la mac de tu tarjeta de red te asignaba una IP, y asociada a esa IP una determinada ruta. Entonces, verificamos que si cambiabamos de tarjeta de red, nuestra latencia respecto a los servidores de quake3, aumentaba o disminuía en función de la ruta que ono había asignado a esa IP. Con los routers sólo cabía esperar que nos cambiasen la IP y con ello la ruta.

Dado que los usuarios del local, no todos son unos expertos informáticos, a veces esto generaba dolores de cabeza (sobre todo a mi) ;), la solución, hacerlo de una manera transparente para ellos.

Los usuarios obtienen su ip por dhcp, siempre la misma, en funcion de la MAC de su tarjeta de red:

```
option routers 172.31.0.254;
option domain-name-servers 62.42.230.135, 62.42.230.136;
host gravis {
hardware ethernet 00:c0:26:26:21:40;
fixed-address 172.31.0.8;
}
```

¿Por qué?

En un principio era para aprovechar las 3 ips públicas, así 6 usuarios simultáneamente podrían conectarse al irc. Pero una vez conseguido este objetivo, ves que es tan fácil, que empiezas a redireccionarlo TODO!!!

¿Qué necesitamos?

Imprescindibles:

Iptables
Ip (paquete iproute2)
Kernel compilado con soporte de target MARK para iptables.

Una vez tenemos todo lo necesario, nos ponemos manos a la obra.

Usaré estas asociaciones para no liar mucho la cosa:

```
IP=/bin/ip
IPTABLES=/sbin/iptables
EXTIF="eth0" # cablemodem (62.42.x.x)
MEDIF="eth1" # hacia router1 (192.168.0.2)
INTIF="eth3" # LAN (172.31.0.254)
MEDIF2="eth2" # hacia router2 (192.168.1.2)
```

Las ips de los routers son: 192.168.0.1 y 192.168.1.1 respectivamente.

Lo que debemos tener claro es por dónde queremos que se vaya cada conexión.

Hay que aclarar que en el local tenemos a jugadores de warcraft 3 y de quake3 considerados de los mejores de España, y si echais un vistazo a la clasificación de war3 2v2 en Europa, vereis que están entre los 10 primeros.Son los del clan Palpelo ;)

Las partidas de warcraft 3 irán siempre por el cable modem, ya que es en el único que puedo hacer NAT. En los routers es mas complicado, seguro que se puede hacer, pero lo quiero hacer con linux.

¿Por qué NAT para warcraft 3? Es necesaria si se quieren hacer equipos, ya que contactan los dos miembros cuando crean un equipo concertado. Gracias a que la gente de Blizzard se lo ha currado un poco, podemos elegir el puerto que queremos que escuche el juego si no nos va bien el tcp 6112 que viene por defecto, pues hala, a hacer NAT:

```
echo " Pasillitos para Warcraft 3 :)"
$IPTABLES -A PREROUTING -t nat -p tcp -i $EXTIF --dport 6112 -j DNAT --to
172.31.0.14:6112
$IPTABLES -A PREROUTING -t nat -p tcp -i $EXTIF --dport 6114 -j DNAT --to
172.31.0.10:6114
$IPTABLES -A PREROUTING -t nat -p tcp -i $EXTIF --dport 6113 -j DNAT --to
172.31.0.112:6113
$IPTABLES -A PREROUTING -t nat -p tcp -i $EXTIF --dport 6115 -j DNAT --to
172.31.0.155:6115
$IPTABLES -A PREROUTING -t nat -p tcp -i $EXTIF --dport 6116 -j DNAT --to
172.31.0.15:6116
$IPTABLES -A PREROUTING -t nat -p tcp -i $EXTIF --dport 6117 -j DNAT --to
172.31.0.12:6117
$IPTABLES -A PREROUTING -t nat -p tcp -i $EXTIF --dport 6120 -j DNAT --to
172.31.0.16:6120
$IPTABLES -A PREROUTING -t nat -p tcp -i $EXTIF --dport 6125 -j DNAT --to
172.31.0.135:6125
$IPTABLES -A PREROUTING -t nat -p tcp -i $EXTIF --dport 6118 -j DNAT --to
172.31.0.18:6118
$IPTABLES -A PREROUTING -t nat -p tcp -i $EXTIF --dport 6119 -j DNAT --to
172.31.0.98:6119
```

```
$IPTABLES -A PREROUTING -t nat -p tcp -i $EXTIF --dport 6121 -j DNAT --to
172.31.0.13:6121
$IPTABLES -A PREROUTING -t nat -p tcp -i $EXTIF --dport 6127 -j DNAT --to
172.31.0.150:6127
$IPTABLES -A PREROUTING -t nat -p tcp -i $EXTIF --dport 6127 -j DNAT --to
172.31.0.120:6128
```

Con esto todos los warcrafteros ya pueden jugar. En este caso no hay que hacer ninguna virguería, se hace con el forwarding básico que no explicaré aquí.

Como no queremos que cuando la peña juega, haya lagazos, todo el tráfico web, mail, ftp, p2p, irc, se ira por los cable routers. Aquí entran en juego ip e iptables.

Lo primero es crear las tablas de rutas para cada gateway:

```
echo 255 local > /etc/iproute2/rt_tables
echo 254 main >> /etc/iproute2/rt_tables
echo 253 default >> /etc/iproute2/rt_tables
echo 0 unspec >> /etc/iproute2/rt_tables
echo 200 router1 >> /etc/iproute2/rt_tables
echo 201 router2 >> /etc/iproute2/rt_tables
echo 202 web >> /etc/iproute2/rt_tables
```

Ya hemos creado unos identificadores para las rutas que irán por los routers. La tabla de web, es para hacer balanceo solo para la web.
Para que las interfaces asociadas a los routers hagan forwarding:

```
$IPTABLES -A FORWARD -i $INTIF -o $MEDIF -j ACCEPT
$IPTABLES -t nat -A POSTROUTING -o $MEDIF -j MASQUERADE
$IPTABLES -A FORWARD -i $INTIF -o $MEDIF2 -j ACCEPT
$IPTABLES -t nat -A POSTROUTING -o $MEDIF2 -j MASQUERADE
```

Ahora asociamos una marca asignada a cada tabla:

```
$IP rule add fwmark 1 table router1
$IP rule add fwmark 2 table router2
$IP rule add fwmark 3 table web
```

El parámetro fwmark es con el cual asignaremos la marca que se incluirá en la cabecera IP del paquete

Ahora debemos asignar las rutas para estas tablas:

```
$IP route add table web eql nexthop via 192.168.1.2 dev $MEDIF2 nexthop via
192.168.0.2 dev $MEDIF
```

Como vemos aquí arriba, se esta haciendo balanceo, entre los dos routers para el tráfico web. Es decir, al no especificar ningún tipo de peso (weight) para cada interfaz las peticiones se repartirán por igual entre las dos interfaces, podemos establecer que una de ellas reciba mas conexiones según nuestras necesidades. Si lo leemos en un idioma inteligible para los seres mortales ;) sería: Añade a la tabla destinada a las conexiones web estas dos puertas de enlace a traves de estas dos tarjetas de red, e intenta igualar el tráfico que le mandas a cada una de ellas por favor (hay que ser educados con nuestro linux ;))

```
$IP route add default via 192.168.0.2 dev $MEDIF table router1
$IP route add default via 192.168.1.2 dev $MEDIF2 table router2
```

Ahora le llega el turno a los paquetes en si, le diremos a nuetro maravilloso linux que las conexiones que le lleguen para los puertos:

20:21 ftp-data:ftp
80 web
6666 irc
6668 irc
110 pop3
25 smtp
22 ssh
4662 p2p
5121 neverwinter nights
2796x quake3 (x=0,1,2)

sean marcadas en función de la puerta de enlace que queremos que usen:

```
$IPTABLES -A PREROUTING -t mangle -i $INTIF -p tcp --dport 80 -j MARK --set-mark
3
$IPTABLES -A PREROUTING -t mangle -i $INTIF -p tcp --dport 20:21 -j MARK
--set-mark 1
$IPTABLES -A PREROUTING -t mangle -i $INTIF -p tcp --dport 22 -j MARK --set-mark
2
$IPTABLES -A PREROUTING -t mangle -i $INTIF -p tcp --dport 25 -j MARK --set-mark
1
$IPTABLES -A PREROUTING -t mangle -i $INTIF -p tcp --dport 6668 -j MARK
--set-mark 1
$IPTABLES -A PREROUTING -t mangle -i $INTIF -p tcp --dport 6666 -j MARK
--set-mark 2
$IPTABLES -A PREROUTING -t mangle -i $INTIF -p udp --dport 27960:27962 -j MARK
--set-mark 2
$IPTABLES -A PREROUTING -t mangle -i $INTIF -p udp --dport 28960 -j MARK
--set-mark 2
$IPTABLES -A PREROUTING -t mangle -i $INTIF -p udp --dport 5121 -j MARK
--set-mark 2
$IPTABLES -A PREROUTING -t mangle -i $INTIF -p tcp --dport 110 -j MARK
--set-mark 1
$IPTABLES -A PREROUTING -t mangle -i $INTIF -p tcp --dport 4662 -j MARK
--set-mark 1
```

Ahora ya tenemos las marcas asignadas :)

Básicamente lo que hace el kernel de nuestro servidor, es analizar los paquetes que le entran por $INTIF (la targeta de red para la LAN) y cuando los paquetes coinciden con los puertos especificados, saltamos al target MARK y los marcamos con el valor especificado, en este caso 1, 2 o 3.

Ahora podemos comprobar que todo funciona correctamente con nuestro amigo iptraf:

Sesion irc:

```
172.31.0.18:1053 = 66 3424 --A- eth3
217.75.226.197:6666 = 83 9431 -PA- eth3
192.168.1.1:1053 = 66 3178 --A- eth2
217.75.226.197:6666 = 83 39545 -PA- eth2
```

Sesion irc:

```
217.75.226.197:6668 > 20 1993 -PA- eth1
192.168.0.1:1034 > 18 776 --A- eth1
217.75.226.197:6668 > 20 1981 -PA- eth3
172.31.0.232:1034 > 18 872 --A- eth3
```

Partida de warcraft:

```
172.31.0.135:2993 > 78 3900 -PA- eth3
213.187.90.245:6112 > 78 3744 --A- eth3
62.42.xxx.xxx:2993 > 78 3744 -PA- eth0
213.187.90.245:6112 > 78 3900 --A- eth0
```

Estos datos corresponden a una captura simultánea, como podemos observar, están activas todas las interfaces al mismo tiempo.

Observaciones:

Para que estas redirecciones, funcionen, en la caso de los routers, hay que desactivar en la interfaces asociadas a ellos, el reverse path filtering:

```
echo "0" > /proc/sys/net/ipv4/conf/eth1/rp_filter
echo "0" > /proc/sys/net/ipv4/conf/eth2/rp_filter
```

Como estamos enrutando hacia dispositivos que también guardan una tabla de estado de las conexiones activas y estos paquetes no siguen una ruta "principal", si no tenemos desactivado este filtro, cuando vuelvan los paquetes, el kernel verá que se han enrutado de otra manera a través de otro gateway y los descartará.

E-mail del autor: metalzone _ARROBA_ teleline.es
Podrás encontrar este artículo e información adicional en: http://bulma.net/body.phtml?nIdNoticia=1615