

IP Aliasing en 1 minuto

Javier Quintano
Documento Personal
javier@jaxvinet.homelinux.org
versión 0.1
12 de noviembre de 2004

IP Aliasing es la técnica que usamos para asignar a una interface física de red tantas direcciones IP como necesitemos. Este documento es sólo un apunte semiordenado de documentación de terceros recopilada y sintetizada aquí junto a algunas prácticas personales.

Copyrighted () 2004 to Javier Quintano - jaXvi - Se otorga permiso para copiar, distribuir y/o modificar este documento bajo las condiciones de la Licencia GNU para Documentación Libre, versión 1.1 o posterior, publicada por la *Free Software Foundation*, sin secciones invariantes. Los fragmentos obtenidos de inet, son propiedad de sus respectivos autores.

Introducción

Juan José Ciarlante realizó una creación poderosa. En 1995, trabajaba en el Departamento General de Irrigación de Mendoza y había conseguido conectarse a una red con Internet, pero debía compartir la conexión con la Universidad Nacional de Cuyo. Tenía una sola placa para dos direcciones IP. Había conocido el software libre y la necesidad de tener dos direcciones en una misma placa me llevó a desarrollar el IP Aliasing, que permite tener un número ilimitado de direcciones sobre una placa, cuenta Ciarlante. Este invento todavía no existe en Windows. Comenzó en agosto del 95 y en octubre su código fue aceptado en Linux, por el mismo Torvalds y su segundo, Alan Cox. Mientras que Windows programa encerrado en sus secretos de Estado, Ciarlante envió su parche al público y recibió mensajes de todo el mundo con correcciones y felicitaciones. Aprendés a programar en grupo, sin tener omnipotencia de decisión, tenés que aprender las reglas del bazaar (feria). Sentía inferioridad antes de enviar el código, por tener una dirección .ar, pero el futuro demostró que la horizontalidad es total, que no importa tu chapa, sino tu código, dice Ciarlante. En ese entonces, Linus Torvalds incorporó el código al kernel de Linux después de haberlo testado entre cientos de personas, aunque estaba en estado embrionario. No existe otro ámbito similar donde tu trabajo pueda ser discutido por colegas tan grosos... Y donde no prime un interés competitivo que evite tal intercambio de ideas, dice Ciarlante, que sigue en Irrigación. Es un proceso evolutivo, cubrí una necesidad que hoy usan miles de personas, asegura. Algunas aplicaciones como Apache (servidores), Sendmail (para enviar correos), Squid (para compartir computadoras), superan el 60% de uso donde compiten. (fuente:inet)

Haciendo IP alias

Hay algunas aplicaciones para las que es útil ser capaz de asignar varias direcciones IP a un sólo dispositivo de red. Los Proveedores de Servicios de Internet usan esto a menudo para ofertar a sus usuarios WWW y FTP «a medida». Puede acudir al IP-Alias mini-HOWTO si quiere obtener más información de la que aquí hay.

Opciones de Compilación del Núcleo:

Networking options --->

```
....
[*] Network aliasing
....
<*> IP: aliasing support
```

Después de compilar e instalar su núcleo con la implementación de IP_Alias, la configuración es muy sencilla. Los alias se añaden a dispositivos de red virtuales asociados al verdadero dispositivo de red. Se aplica una convención sencilla para dar nombres a estos dispositivos, que es <nombredisp>:<núm disp virtual>, por ejemplo eth0:0, ppp0:10, etc. Tenga en cuenta que dispositivo nombre:número sólo se puede configurar después de haber activado la interfaz principal.

Por ejemplo, asumiremos que tiene una red Ethernet que soporta dos redes IP diferentes simultáneamente y quiere que tu máquina tenga acceso directo a ambas. Podríamos hacer algo como:

```
root# ifconfig eth0 192.168.1.1 netmask 255.255.255.0 up
root# route add -net 192.168.1.0 netmask 255.255.255.0 eth0
root# ifconfig eth0:0 192.168.10.1 netmask 255.255.255.0 up
root# route add -net 192.168.10.0 netmask 255.255.255.0 eth0:0
```

Para borrar un alias sencillamente añadimos un '-' al final de su nombre y nos referimos a él de forma tan sencilla como:

```
root# ifconfig eth0:0- 0
```

También serán borradas automáticamente todas las rutas asociadas con ese alias.

Ejemplo práctico

Disponemos de un escenario con los siguientes host:

Ordenador portátil:

```
portable:/home/jaxvi# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:C0:9F:24:FC:CB
          inet addr:192.168.0.5  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:46045 errors:0 dropped:0 overruns:0 frame:0
          TX packets:86202 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5083106 (4.8 MiB)  TX bytes:120473462 (114.8 MiB)
          Interrupt:11 Base address:0xe200
```

```

lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:86 errors:0 dropped:0 overruns:0 frame:0
            TX packets:86 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:6676 (6.5 KiB)  TX bytes:6676 (6.5 KiB)

```

Ordenador personal:

```

torre-02:~# ifconfig
eth0       Link encap:Ethernet  HWaddr 00:01:80:4B:DE:E0
            inet addr:192.168.0.2  Bcast:192.168.0.255  Mask:255.255.255.0
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:93848 errors:0 dropped:0 overruns:0 frame:0
            TX packets:50765 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:126421568 (120.5 MiB)  TX bytes:4099500 (3.9 MiB)
            Interrupt:18 Base address:0xd000

lo         Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:192913 errors:0 dropped:0 overruns:0 frame:0
            TX packets:192913 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:32708141 (31.1 MiB)  TX bytes:32708141 (31.1 MiB)

```

Como vemos ambos pertenecen a la misma red, se pingean entre si, s n necesidad de a adir ni configurar rutas por defecto etc ...

Haciendo un repaso r pido recordemos algunas configuraciones t picas para rangos de IP en LAN:

Clase	Valor de w	Id. de red	Id. de host	N�mero de redes	N�mero de hosts por red		
A	1-126	w	x.y.z	126	16,777,214		
B	128-191	w.x	y.z	16,384	65,534		
C	192-223	w.x.y	z	2,097,152	254		
D	224-239	Reservado para direcciones de multidifusi�n			No disponible	No disponible	No disponible
E	240-254	Reservado para uso experimental			No disponible	No disponible	No disponible

tengamos en cuenta los rangos reservados para IP internas dentro de las clases, que son:

```

192.168.x.x para la clase C
172.16.x.x para la clase B
10.x.x.x para la clase A

```

Podemos probar a a adir IP privadas de clase A a nuestros hosts y ver si se responden. Para esto levantamos un alias en la m quina Torre-02:

```

torre-02:~# ifconfig eth0:0 10.10.10.2
torre-02:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:01:80:4B:DE:E0
          inet addr:192.168.0.2  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:94057 errors:0 dropped:0 overruns:0 frame:0
          TX packets:50873 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:126464498 (120.6 MiB)  TX bytes:4113065 (3.9 MiB)
          Interrupt:18 Base address:0xd000

eth0:0    Link encap:Ethernet  HWaddr 00:01:80:4B:DE:E0
          inet addr:10.10.10.2  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:18 Base address:0xd000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:212469 errors:0 dropped:0 overruns:0 frame:0
          TX packets:212469 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:36000898 (34.3 MiB)  TX bytes:36000898 (34.3 MiB)

```

```

torre-02:~# route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
192.168.0.0     0.0.0.0         255.255.255.0  U      0      0      0 eth0
10.0.0.0        0.0.0.0         255.0.0.0      U      0      0      0 eth0
0.0.0.0         192.168.0.1    0.0.0.0        UG     0      0      0 eth0

```

En el portátil intentamos pingear a esa IP sin resultados, posteriormente levantamos un alias para esa dirección de red y comprobamos que responde, en ambos casos no ha hecho falta levantar ninguna ruta, ya que no estamos intentando alcanzar ninguna red remota.

```

portable:/home/jaxvi# ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2): 56 data bytes

--- 10.10.10.2 ping statistics ---
2 packets transmitted, 0 packets received, 100% packet loss
portable:/home/jaxvi# ifconfig eth0:0 10.10.10.5
portable:/home/jaxvi# ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2): 56 data bytes
64 bytes from 10.10.10.2: icmp_seq=0 ttl=64 time=1.4 ms
64 bytes from 10.10.10.2: icmp_seq=1 ttl=64 time=0.2 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=64 time=0.2 ms

--- 10.10.10.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.6/1.4 ms

```

Hasta aquí nada interesante. De hecho yo aún no he encontrado una utilidad real, tendré que mirar más ...
:s

En cambio al menos me ha servido para comprobar el buen funcionamiento de algunas de las reglas de filtrado del firewall de mi GW.

Realicé la misma operación en un tercer host, corriendo iptables, añadí una interface virtual, del rango anterior:

```
eth1:0    Link encap:Ethernet  HWaddr 00:02:44:6F:58:92
          inet addr:10.10.10.1  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:10  Base address:0x1400
```

y a continuación el ping de rigor ;)

```
portable:~# ping 10.10.10.1
PING 10.10.10.1 (10.10.10.1): 56 data bytes

--- 10.10.10.1 ping statistics ---
 3 packets transmitted, 0 packets received, 100% packet loss
```

Efectivamente, ni siquiera un ping debería colarse desde dentro de la LAN hacia fuera (mucho menos al revés), si no está en mi red de confianza.

```
iptables -A INPUT -i eth1 -s 192.168.0.0/24 -j ACCEPT
```

Podríamos probar a jugar con el firewall y permitir los ping etc ..., ahora ya tenemos varias redes para hacer pruebas de manera sencilla y limpia sólomente añadiendo alias :)

Por último compruebo, si los servicios que están a la escucha en una interface, lo siguen estando en las nuevas interfaces o alias que creamos.

```
portable:~# telnet 10.10.10.2 631
Trying 10.10.10.2...
Connected to 10.10.10.2.
Escape character is '^]'.
get farsa
HTTP/0.9 405 Unknown
Date: Sat, 13 Nov 2004 23:37:37 GMT
Server: CUPS/1.1
Allow: GET, HEAD, OPTIONS, POST
Upgrade: TLS/1.0,HTTP/1.1
Connection: close
Content-Type: text/html
Content-Length: 89
```

Pues sí!, parece que nuestro server de impresión nos responde :P

.... la verdad se me empiezan a ocurrir varias cosas divertidas con estos de los alias ... pero eso para otro momento...

salu2