



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

Como añadir conectividad IPv6 a Debian GNU/Linux (17373 lectures)

Per **Alberto Rodríguez Galdo**, [matosinho](#) ()

Creado el 11/08/2003 20:32 modificado el 11/08/2003 20:32

IPv6 es el siguiente paso en la evolución de Internet. Es una evolución del protocolo ip (actualmente en su versión 4, IPv4) que aporta mejoras significativas, tales como, mejoras en el enrutamiento, un mayor rango de direcciones ip, mejoras en la autoconfiguración, etc...

En este artículo se explica como añadir conectividad a una máquina con Debian instalado, de manera que podamos comunicarnos con las islas ipv6 actuales a través de un túnel 6to4, y finalmente como activar dicho ordenador para que ofrezca conectividad ipv6 a otros Linux (u otros sistemas operativos con soporte IPv6, solo he comprobado windows xp) conectados a él, actuando como router.

En este artículo conseguirá proveer de conectividad ipv6 a su Linux, de manera que será capaz tanto de conectar a actuales hosts que soportan comunicacion ipv6 como proveer de conectividad ipv6 a otras máquinas Linux conectadas a la suya.

Antes de comenzar, hagamos un repaso de que es lo necesario para completar las instrucciones de este artículo:

- Debian ;) (otra distribución también valdrá)
- Un kernel de la serie 2.4.x con ipv6 activado, enlazado en el kernel o como módulo
- El paquete iproute
- El paquete iputils-ping y el paquete iputils-tracepath
- El paquete radvd

Este artículo estará basado en Debian y una configuración de dos ethernet, una conectada directamente a un modem-cable con una ip pública y otra ethernet conectada a la red local con una ip privada 192.168.0.1. Los cambios para utilizar otra distribución son mínimos y la mayor parte es aplicable sin cambios.

Debemos disponer de un kernel con soporte ipv6, este soporte se encuentra de serie en los kernel 2.4.x en la sección "Networking Options", la opción es "The IPv6 protocol (EXPERIMENTAL)". Como podeis ver es una opción experimental, así que debéis permitir código o drivers experimentales en vuestro kernel, esto se consigue activando "Prompt for development and/or incomplete code/drivers" en la opción "Code maturity level options".

Una vez compilado el kernel con soporte ipv6, reiniciamos con este nuevo kernel y comprobamos que se ha activado este soporte:

```
ifconfig lo
```

Si el kernel esta bien configurado, debajo de la dirección ipv4 127.0.0.1, aparecerá su nueva dirección ipv6:

```
Inet6 addr: ::1/128 Scope:Host
```

Si esto es así, estamos en el buen camino, si no aparece la línea anterior, debemos comprobar que estamos



arrancando con el kernel apropiado, en caso de que hayamos compilado ipv6 como módulo, debemos cargar el módulo de ipv6:

```
modprobe ipv6
```

Ahora instalaremos los paquetes que necesitaremos a lo largo del artículo, iputils-ping e iputils-tracepath, para comprobar que ipv6 funciona y en el caso de que queramos configurar la máquina como gateway para proveer conectividad ipv6 a las máquinas de la red local instalaremos el paquete radvd.

Los paquetes se instalan utilizando apt-get de la siguiente manera:

```
apt-get install iputils-ping
apt-get install iputils-tracepath
apt-get install radvd
```

Ahora que tenemos conectividad ipv6 vamos a conseguir conectarnos con máquinas que se comuniquen con nosotros usando ipv6 a través de Internet. Para esto vamos a utilizar un tipo de túneles especiales, son los túneles 6to4. Un túnel 6to4 utiliza una máquina que se encarga de entender un tipo especial de paquetes ipv4 que son los paquetes 6to4, estos paquetes encapsulan paquetes ipv6 en paquetes ipv4 y crean un túnel entre nosotros y los sitios que entienden ipv6. Esto es necesario por que los ISP todavía no ofrecen conectividad ipv6 por lo que por ahora tenemos que apañármolas con túneles :(.

6to4 utiliza un tipo especial de direcciones, una dirección que traduce nuestra dirección pública única ipv4 en una dirección ipv6 única que nos permitirá conectarnos con el broker 6to4 para tener conectividad ipv6. La dirección 6to4 que nos corresponde se forma utilizando el prefijo 2002: y añadiéndole cada uno de los bytes de nuestra dirección ip ipv4 de dos en dos, en hexadecimal y separados por dos puntos (:). Por ejemplo:

Si mi ip pública es 213.60.102.104, 213 es d5 en hex., 60 es 3c en hex, 102 es 66 en hex. y 104 es 68 en hex., por lo que la dirección 6to4 de 213.60.102.104 es 2002:d53c:6668::

Este es un script bash que nos ayuda a calcular direcciones ipv6 6to4 a partir de direcciones ipv4:

```
printf "2002:%02x%02x:%02x%02x::\n" a b c d
```

Sustituyendo a,b,c,d, por los bytes de nuestra dirección ipv4 pública.

Bien, ahora que tenemos nuestra dirección 6to4 pública, vamos a configurar el túnel. Para esto tenemos que editar el archivo /etc/network/interfaces, añadiendo las siguientes líneas y utilizando la ip que hemos calculado a partir de la ip publica de nuestra interfaz conectada a internet:

```
auto SeisACuatro
iface SeisACuatro inet6 v4tunnel
    address 2002:d53c:6668::1
    netmask 16
    endpoint any local 213.60.102.104
    up ip -6 route add 2000::/3 via ::192.88.99.1 dev SeisACuatro
    metric 1
    down ip -6 route flush dev SeisACuatro
```

Sustituyendo 2002:d53c:6668::1 por cualesquiera que sea la ip que nos corresponda.

Un breve comentario de que significa cada una de las lineas:

Creamos el tunel SeisACuatro que encaminará nuestros paquetes 6to4 al broker, le asignamos al túnel nuestra flamante dirección 6to4, y enrutamos cualquier dirección del rango 2000::/3 a través de la ip anycast del router que nos encaminará al router 6to4 mas cercano a nosotros (::192.88.99.1). En caso de desconfigurar el tunel, hacemos un flush de las rutas añadidas.

Pues bien, una vez añadidas estas líneas al archivo interfaces, ya podemos poner nuestro túnel en funcionamiento:



```
ifup SeisACuatro
```

Ahora comprobamos que funciona correctamente, haciendo un ping a algunas direcciones ipv6 (utilizamos ping6, del paquete iputils-ping). por ejemplo:

```
ping6 -n www.kame.net
ping6 -n www.ipv6.elmundo.es
```

La respuesta debe ser algo parecida a esto:

```
PING www.ipv6.elmundo.es (2001:450:9:10::71) from 2002:d53c:6668::1 : 56 data bytes
64 bytes from 2001:450:9:10::71: icmp_seq=1 ttl=57 time=262 ms
64 bytes from 2001:450:9:10::71: icmp_seq=1 ttl=57 time=263 ms
64 bytes from 2001:450:9:10::71: icmp_seq=1 ttl=57 time=261 ms
```

Como crear el tunel para usuarios de otras distribuciones

El script anterior está orientado, como el resto del artículo a una distribución Debian, si no dispones de una Debian, también puedes crear el túnel utilizando únicamente la utilidad ip del paquete iproute, estas son los comandos (también los puedes utilizar en Debian):

```
ip tunnel add SeisACuatro mode sit ttl 0 remote any local 213.60.102.104
ip link set dev SeisACuatro up
ip -6 addr add 2002:d53c:6668::/16 dev SeisACuatro
ip -6 route add 2000::/3 via ::192.88.99.1 dev SeisACuatro metric 1
```

¡Enhorabuena!, ya puedes conectarte con la futura internet, ya tienes conectividad ipv6 hacia internet. Si no tienes una red interna o no quieres ofrecer conectividad ipv6 a las máquinas de tu red local el artículo termina aquí, pero si quieres ofrecer conectividad ipv6 a las máquinas de tu red local utilizando la máquina que acabas de configurar como gateway, te invito a que sigas leyendo este artículo.

Ahora vamos a configurar la máquina con conectividad ipv6 hacia internet para que sirva de gateway al resto de máquinas de tu red local.

Para esto necesitamos disponer de una ip local ipv6 asignada a la interfaz ethernet que te conecta con tu red local y de un prefijo con el cual el núcleo ipv6 podrá formar direcciones ipv6 automáticamente. Este es uno de los avances de ipv6, la capacidad de auto-configuración.

Si te fijas, una vez que has activado ipv6 en el kernel de Linux, todas tus interfaces que ya estaban configuradas para ipv4, automáticamente adquieren una ip ipv6 en el rango fe80:, estas son las denominadas ip's link-local, son ips locales de tu subred y se forman con el prefijo fe80: y los datos de la MAC de la interfaz de red.

Pues bien, el núcleo de ipv6 utilizará esta direccion link-local para comunicarse con routers ipv6, que cada cierto tiempo, estos routers, le comunicarán el prefijo de su subred y la máquina decidirá que ip se autoconfigura a partir del prefijo y de su propia MAC.

En el caso que nos ocupa, instalaremos un demonio que ejecutará este proceso, denominado "router advertisement" y utilizará una subred a partir de la ip 6to4 que hemos utilizado antes.

Debemos asignarle una ip ipv6 a la interfaz que nos conecta con nuestra red local en la máquina que tiene la ip pública y que actuará como gateway. Para eso editamos el archivo /etc/network/interfaces para añadirle las siguientes líneas (asumiendo que la interfaz ethernet que nos conecta a nuestra red local es eth0):



```
iface eth0 inet6 static
    address 2002:d53c:6668:1::1
    netmask 64
```

Sustituyendo 2002:d53c:6668:1::1 por al ip que nos corresponde a partir de la ip global 6to4.

Le estamos añadiendo la ip 2002:d53c:6668:1::1 con 64 bits de máscara a la interfaz eth0 que nos comunica con la red local.

Para que la nueva configuración surta efecto, ejecutamos:

```
ifdown eth0
ifup eth0
```

Acabamos de asignar una ip a nuestra interfaz gateway dentro de un rango de 64 bits.

Ahora debemos configurar el radvd (router advertisement daemon) para que asigne el prefijo de nuestra subred a todos los nodos que cuelgan del gateway. Para esto, modificamos el archivo /etc/radvd.conf (sustituimos cualquier línea que ya existiese en el archivo) añadiendo las siguientes líneas:

```
interface eth0
{
    AdvSendAdvert on;
    MaxRtrAdvInterval 4;
    MinRtrAdvInterval 3;
    prefix 2002:d53c:6668:1::/64
    {
    };
};
```

Y reiniciamos el demonio, con:

```
/etc/init.d/./radvd restart
```

Ahora una vez reiniciado el demonio, este acaba de anunciarse en la red y ha provocado que todos los host que están conectados a la subred de la interfaz eth0 que tienen ipv6 activado se autoconfiguren una ip dentro de la subred 2002:d53c:6668:1::/64, a la que pertenece nuestro gateway. Los valores de MaxRtrAdvInterval (el valor máximo en segundos entre envíos de paquetes "router advertisement") y MinRtrAdvInterval (tiempo mínimo en segundos para el envío de un paquete "router advertisement") son los mínimos posibles. AdvSendAdvert on, establece que se envíen paquetes "router advertisement" bajo solicitud por parte de uno de los nodos o automáticamente cada cierto tiempo.

Una vez hecho esto, y si no se ha producido ningún error, ya debemos tener conectividad ipv6 en las máquinas de nuestra red local a través del gateway que configuramos en la primera parte de este artículo.

IPv6 es totalmente funcional en este mismo momento, y en los próximos años se irá efectuando una migración progresiva hacia el protocolo ipv6 por las ventajas que aporta. Este artículo nos ofrece la posibilidad de situarnos a la vanguardia de internet y anticiparnos a un futuro no tan lejano.

Enlaces

<http://people.debian.org/~csmall/ipv6/>⁽¹⁾

<http://www.ipv6-es.com>⁽²⁾

http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/html_single/Linux+IPv6-HOWTO.html⁽³⁾

Alberto rodriguez Galdo, Agosto 2003

Este artículo se ajusta a los términos de la licencia [Free Documentation License](#)⁽⁴⁾

Lista de enlaces de este artículo:



1. <http://people.debian.org/~csmall/ipv6/>
2. <http://www.ipv6-es.com>
3. http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/html_single/Linux+IPv6
4. <http://www.gnu.org/copyleft/fdl.html>

E-mail del autor: tetoro _ARROBA_ teleline.es

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=1840>