IPv6 con 6to4 detras de NAT con IPv4 Publica Variable

Carlos Martinez-Cagnazzo

23 de diciembre de 2008

Resumen

6to4 es uno de los varios mecanismos de transicion de IPv4 a IPv6 que han sido propuestos por el IETF. Es un mecanismo de tunelizacion de IPv6 en un payload IPv4 con la caracteristica adicional de que el extremo remoto del tunel puede configurarse automaticamente ya que usa una direccion well-known como extremo remoto. Es facil de configurar y tiene buena performance. Sin embargo, presenta algunos desafios a la hora de poderlo utilizar de manera estable detras de un NAT con IP publica variable, como es el caso de la inmensa mayoria de servicios ADSL hogarenos.

En este articulo presento un *script* que automatiza la configuracion y mantiene operativa la conexion IPv6 utilizando 6to4 utilizando un PC Linux como *router* IPv6.

1. Generalidades de 6to4

6to4 es uno de los varios mecanismos de transicion de IPv4 a IPv6 que han sido propuestos por el IETF. Es un mecanismo de tunelizacion de IPv6 en un payload IPv4 con la caracteristica adicional de que el extremo remoto del tunel puede configurarse automaticamente ya que usa una direccion de terminacion remota bien conocida (192.88.99.1.)

Desde el punto de vista del enrutamiento, 6to4 mapea el espacio de direcciones IPv4 dentro de un prefijo IPv6 (el 2002::/16). De esta forma, toda direccion IPv4 existente en Internet tiene una direccion dentro del prefijo de 6to4 asignada.

2. Caso de estudio

Analizamos el uso de 6to4 en una red como la de la figura 1. Este es un caso bastante típico actualmente, en el cual tenemos una red LAN con numeración privada (asumimos 192.168.1.0/24 para el caso de estudio), algunas PCs de trabajo que pueden ser Windows o Linux indistintamente, y una caja Linux, la cual si bien en principio puede correr cualquier distribución, en el resto de este paper asumiremos es Ubuntu o Debian.

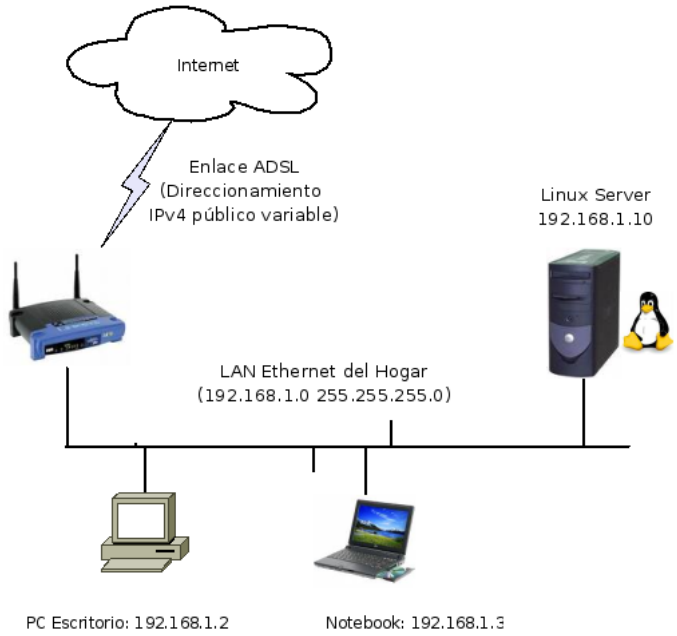


Figura 1: Red caso de estudio

La conectividad a Internet esta provista por un servicio ADSL y un router hogareño que conecta la LAN a Internet con una de direccion IP dinámica, realizando las funciones de tunelizacion de *PPP over Ethernet* (PPPoE) y NAT para los computadores de la LAN privada.

3. Configuración manual de 6to4

Antes de analizar como automatizar veremos como configurar manualmente 6to4 en Linux. Los pasos que deberemos seguir son:

1. Calcular nuestra dirección IPv6
2. Configurar la nueva interfaz IPv6 y la ruta por defecto
3. Verificar el funcionamiento

3.1. Calcular nuestra dirección IPv6

En 6to4 nuestra dirección IPv6 se deriva a partir de nuestra dirección pública IPv4. Si por ejemplo, nuestra **IPV4=1.2.3.4** entonces nuestra IPV6 será **IPV6=2002.0102.0304::1**, asumiendo que decidamos utilizar la dirección "1" para nuestro host 6to4. Esto no es obligatorio pero es recomendado.

En Unix es posible automatizar la generación de nuestra dirección IPv6 de la siguiente manera ([1]):

```
ipv4="1.2.3.4";
printf "2002:%02x%02x:%02x%02x::1" $(echo $ipv4 | tr "." " ")
```

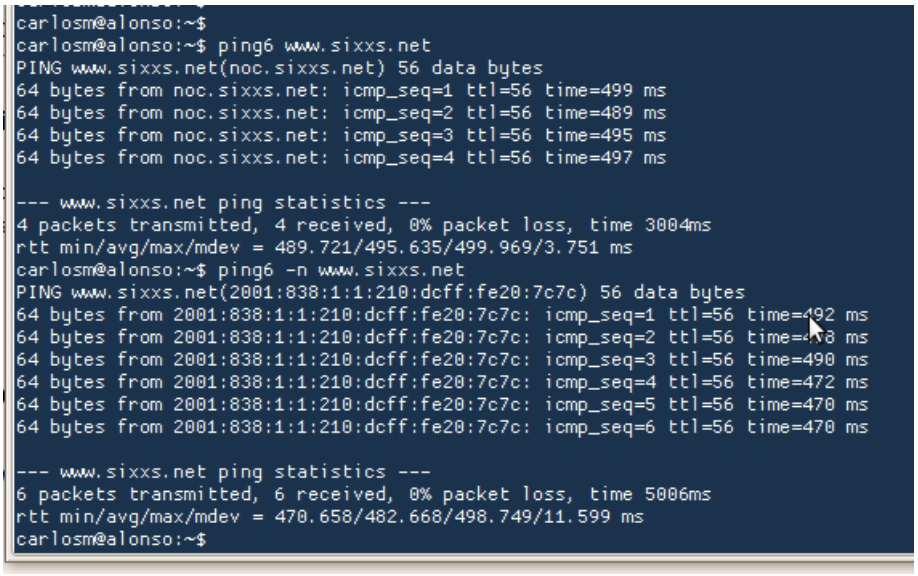
3.2. Configurar la nueva interfaz IPv6 y la ruta por defecto

Luego de determinada nuestra dirección IPv6, debemos proceder a configurar la interfaz de túnel, siguiendo los pasos de acuerdo a [1]:

Crar la interfaz de túnel
`# /sbin/ip tunnel add tun6to4 mode sit ttl <ttldefault>remote any local <localipv4address>`
Levantar la interfaz
`# /sbin/ip link set dev tun6to4 up`
Agregar la dirección IPv6 calculada a la interfaz. Notar que el largo del prefijo es /16
`# /sbin/ip -6 addr add <local6to4address>/16 dev tun6to4`
Agregar una ruta por defecto utilizando la ruta "bien conocida" a todo el prefijo 6to4:
/sbin/ip -6 route add 2000::/3 via ::192.88.99.1 dev tun6to4 metric 1
o
/sbin/ip -6 route add 2000::/3 via 2002:c058:6301::1 dev tun6to4 metric 1

3.3. Verificar el funcionamiento

Una vez realizados todos estos pasos, deberíamos tener ya conectividad IPv6. Para verificar, podemos utilizar cualquiera de las clasicas herramientas ping, traceroute o wget en sus formas IPv6 (ping6, traceroute6 o wget -6)



4. Automatizacion de la configuracion de 6to4

De los pasos de configuración mostrados anteriormente puede verse que la conectividad IPv6 usando 6to4 fallará cuando la dirección v4 pública de nuestro router ADSL cambie. En esta sección se presenta un script en shell que automatiza todos los pasos de configuración anteriores y además verifica la dirección IPv4 pública de nuestro router ADSL y en caso de ser necesario, reconfigura el 6to4 para mantener nuestra conectividad IPv6 activa.

```
1  #! /bin/sh
2  # /etc/init.d/6to4
3  #
       ###################################################################
4  # 6to4-linux-debian.sh : Script to start / stop 6to4 IPv6
5  # routing with provisions for a "behind dynamic ip w/NAT"
6  # setup and interoperation with radvd router advertising daemon.
7  # Version 0.2
8  #
9  # (c) Carlos M. Martinez-Cagnazzo, 12-dec-2008
10 #
11 # Changelog:
12 # Version 0.3 (20081229) : added start, stop, restart functions
13 #                          Flush is necessary for correct restart
14 #                          Adds list of neighbors to status
15 # Version 0.2 (20081212) :      first "usable" version, converted
       to "init"
16 #                               script format
17 #
       ###################################################################

18
19 # Some things that run always
20 touch /var/lock/6to4
21 ipfile="/var/tmp/ipv4.txt"
22
23 # implements startup of 6to4
24 start() {
25         # Check if there is old information about public IPv4 addr
               available
26         if [ -f $ipfile ]; then
27                 ipv4_last=$(cat /tmp/ip.txt)
28         fi
29         # Need to get my ip automatically. I use whatismyip.com's
               automation page
30         wget --output-document=$ipfile http://whatismyip.com/
               automation/n09230945.asp
31         ipv4_int="192.168.1.10"
32         ipv4_cur=$(cat $ipfile)
33         ipv6=$(printf "2002:%02x%02x:%02x%02x::1" $(echo $ipv4_cur
               | tr "." " "))
34
35         # beware: the "ffff" part must be coherent with the
               settings in /etc/raddv.conf
36         lan_prefix=$(printf "2002:%02x%02x:%02x%02x:ffff::/64" $(
               echo $ipv4_cur | tr "." " "))
37         eth0_ipv6=$(printf "2002:%02x%02x:%02x%02x:ffff::1/64" $(
               echo $ipv4_cur | tr "." " "))
38
39         # display status
40         echo My PUBLIC IPv4 address is $ipv4_cur
```

```
41          echo My GLOBAL IPv6 address is $ipv6
42          echo My LAN PREFIX is $lan_prefix
43          # echo Configuring...
44          ip tunnel add tun6to4 mode sit ttl 200 remote any local
                $ipv4_int
45          ip link set dev tun6to4 up
46          ip -6 addr add $ipv6/48 dev tun6to4
47          ip -6 addr add $eth0_ipv6 dev eth0
48          ip -4 addr add $ipv4_cur dev tun6to4
49          ip -6 route add 2000::/3 via ::192.88.99.1 dev tun6to4
                metric 1
50          # ip -6 route add ::/0 via ::192.88.99.1 dev tun6to4 metric
                1
51          # Make raddvd's advertised prefix routeable through eth0
52          ip -6 route add $lan_prefix dev eth0 metric 1
53          # Restart RADVD if necessary
54          if [ -x /etc/init.d/radvd ]; then
55                  /etc/init.d/radvd restart
56          fi
57          # log the realized operation
58          logger "IPv6 using 6to4 configured (ipv4_pub: $ipv4_cur,
                ipv6: $ipv6)"
59  }
60
61  # stops 6to4 and flushes all routes and addresses
62  stop() {
63          /sbin/ip -6 route flush dev tun6to4
64          /sbin/ip -6 route flush scope global
65          /sbin/ip -6 addr flush scope global
66          /sbin/ip link set dev tun6to4 down
67          /sbin/ip tunnel del tun6to4
68  }
69
70  status() {
71      echo "IPV6 6TO4 STATUS"
72      echo "__"
73          echo "Addresses"
74          echo "_________"
75          /sbin/ip -6 addr show
76          echo " "
77          echo "Routes"
78          echo "______"
79          /sbin/ip -6 route show
80          echo "Neighbors"
81          echo "______"
82          /sbin/ip -6 neigh show
83  }
84
85  restart() {
86          stop
87          start
88  }
89
90  # Carry out specific functions when asked to by the system
91  case "$1" in
92    start)
93      echo "Starting script 6to4 "
94          start
95      ;;
96    stop)
97      echo "Stopping script 6to4"
98          stop
```

```
 99      ;;
100    status)
101            status
102      ;;
103    restart)
104            restart
105            ;;
106    *)
107      echo "Usage: /etc/init.d/6to4 {start|stop|restart|status}"
108      exit 1
109      ;;
110  esac
111
112  exit 0
```

5. Conectando al resto de la LAN con "rdadv"

Utilizando *stateless autoconfiguration* es posible dar conectividad IPv6 al resto de los computadores de la red LAN del caso de estudio.

6. Referencias

Referencias

[1] P. Bieringer, “Linux ipv6 howto,” 10 2007.