Un tunnel IPv6-in-v4 sur un tunnel GRE...

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 3 septembre 2011. Dernière mise à jour le 14 septembre 2011

https://www.bortzmeyer.org/tunnel-over-tunnel.html

Si vous voulez connecter un réseau local à l'Internet IPv6, mais que vous êtes coincé derrière un FAI qui ne fournit encore, en 2011, qu'IPv4, que faire ? La solution évidente est de configurer un tunnel vers un fournisseur de connectivité IPv6. Ce cas est bien connu et on trouve d'innombrables HOWTO et articles de blog sur ce sujet. Mais, que faire si le routeur d'entrée du site n'a pas IPv6, et que la machine qui pourrait servir de routeur IPv6 n'a pas d'adresse IPv4 publique ? Simple, il suffit de créer un deuxième tunnel, entre le routeur d'entrée de site et le routeur IPv6, et de faire tourner le second tunnel sur le premier...

Le cas réel m'est arrivé à l'université de Yaoundé 1 pour une formation sur IPv6. Le Cameroun a pour l'instant zéro connectivité IPv6. L'université n'a que peu d'adresses IPv4 publiques mais il y en avait quand même une de disponible pour établir un tunnel « IPv6 dans IPv4 » (protocole 41) avec un fournisseur de connectivité IPv6 comme l'excellent Hurricane Electric <https://www.bortzmeyer.org/ipv6-he.html>. Le seul problème est que, pour des raisons matérielles, le réseau où se trouvaient les machines à adresse publique ne permettait pas de nouvelle installation. Dans ce réseau, un routeur Cisco 2800 aurait pu servir de routeur IPv6 (l'excellente interface Web de Hurricane Electric <http://www.tunnelbroker.net/> fournit des instructions IOS qu'on n'a qu'à copier/coller ou bien on peut utiliser ce tutoriel <https://learningnetwork.cisco.com/docs/DOC-10847>). Mais ce Cisco-là n'avait pas IPv6, protocole qui nécessite une licence spéciale (baptisée "*Advanced*" quelque chose) et une image d'IOS nouvelle. Après une heure passée à naviguer dans les options incompréhensibles du site Web de Cisco, pour essayer de trouver un moyen de mettre à jour la machine, nous avons renoncé.

Bon, il faut donc mettre le routeur IPv6 sur un PC Debian. Il y a encore plus de documentation pour ce système, et cela a l'avantage d'utiliser du logiciel libre. Mais il y avait des limites matérielles : pas de PC avec deux cartes Ethernet disponible, pas de possibilité de le mettre dans la salle machines, etc, qui ont mené à une deuxième décision : le routeur IPv6 sur PC reste dans les salles où se trouvent les autres PC de l'université, et on va créer un tunnel entre lui et le Cisco, « apportant » l'adresse IPv4 publique dont a besoin Hurricane Electric jusqu'à ce PC.

Donc, les acteurs vont être :

— Hurricane Electric (POP de Londres, le plus proche), adresse IPv4 `216.66.80.26` et IPv6 `2001:470:1f08:1c`
— Le routeur IPv6 de l'Université, un PC Debian/Linux, adresse IPv4 sur le réseau local `172.16.0.76` et adresse IPv4 publique`41.204.94.222`, en IPv6, `2001:470:1f08:1c8f::2`. Le réseau alloué à l'unversité de Yaoundé est `2001:470:1f09:1c8f::/64` (avec les adresses IPv6, ouvrez l'œil : 1f08 n'est pas 1f09).
— Le Cisco 2800 situé à l'entrée de l'Université, adresse IPv4 `41.204.93.101`.

Le problème est donc de permettre à la machine d'HE (Hurricane Electric) d'atteindre `41.204.94.222`. J'ai choisi un tunnel GRE (RFC 2784 [1]) pour sa simplicité, et son caractère normalisé (tout le monde sait faire du GRE). Si le tunnel, au lieu d'être limité à l'université, s'était étendu sur un réseau public, il aurait été plus prudent de le protéger par la cryptographie. Mes "*followers*" sur Twitter ont voté massivement pour OpenVPN mais, ici, GRE suffit.

Pour l'intérieur du tunnel GRE, j'ai choisi les adresses IP `192.0.2.1` (pour le Cisco) et `192.0.2.2` (pour le PC). Attention si vous tentez une expérience similaire : il faut vraiment y aller par étapes, en testant soigneusement le résultat à chaque étape. Autrement, le déboguage est cauchemardesque.

On va d'abord dire au Cisco de créer son côté du tunnel GRE, sur l'interface `Tunnel0` :

```
int tunnel 0
ip address 192.0.2.1 255.255.255.0
tunnel source 41.204.93.101
tunnel destination 172.16.0.76
exit

ip route 41.204.94.222/32 tunnel0
```

Ainsi, IOS envoit désormais les paquets qu'il reçoit pour `41.204.94.222` à `172.16.0.76`, en GRE. Côté du PC destinataire, on va configurer l'interface `tun0` :

```
ip tunnel add tun0 mode gre remote 41.204.93.101 local 172.16.0.76 ttl 255
ip link set tun0 up
ip addr add 192.0.2.2 dev tun0
ip route add 192.0.2.1/24 dev tun0
```

puis mettre l'adresse IPv4 publique sur une interface `dummy0` :

```
ifconfig dummy0  up
ifconfig dummy0  41.204.94.222/32
```

Attention, avec ce système, le routage est asymétrique. Un paquet entrant passera par l'interface `tun0` alors que la réponse sortira par le normal `eth0`. On peut changer la table de routage pour forcer le passage par le tunnel GRE, ou bien on peut tout simplement débrayer le contrôle du routage asymétrique (qui est activé par défaut sur Debian, pour limiter les risques d'usurpation d'adresses IP) en mettant les variables sysctl `net.ipv4.conf.all.rp_filter` et `net.ipv4.conf.default.rp_filter` à 0. Autrement, vous verrez les paquets être mystérieusement jetés par Linux.

Et voilà, le tunnel GRE fonctionne. Depuis tout l'Internet, on peut pinguer `41.204.94.222`, bien que la machine soit reléguée loin du routeur d'entrée, dans une zone uniquement RFC 1918. Un autre avantage de GRE est que tcpdump le connait et l'analyse très bien, rendant bien plus facile le déboguage. Voici un exemple de paquet qui passe par le tunnel. On a capturé le trafic sur l'interface Ethernet et on voit les paquets GRE entre `41.204.93.101` (le routeur Cisco) et `172.16.1.2` (le PC à l'autre bout du tunnel GRE). Ici, le trafic dans le tunnel était une connexion SMTP depuis `66.216.133.144` vers `41.204.94.222` :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc2784.txt`

```
12:11:21.894937 IP 41.204.93.101 > 172.16.1.2: GREv0, length 56: \
   IP 66.216.133.144.26892 > 41.204.94.222.25: \
        Flags [S], seq 1940136218, win 8192, options [mss 1380,nop,wscale 8,nop,nop,sackOK], length 0
```

Via l'interface Web d'Hurricane Electric (qui teste la connectivité effective avec l'adresse donnée, donc la configuration préalable de GRE était nécessaire), on peut créer le tunnel IPv6-dans-IPv4. Reste à le configurer sur le PC (le Cisco n'a plus rien à faire, il laissera passer ce qu'il prend pour des paquets IPv4 normaux).

Les instructions de HE sont simples, on utilise une interface nommée `he-ipv6` :

```
ip tunnel add he-ipv6 mode sit remote 216.66.80.26 local 41.204.94.222 ttl 255
ip link set he-ipv6 up
ip addr add 2001:470:1f08:1c8f::2/64 dev he-ipv6
ip route add ::/0 dev he-ipv6
```

La première ligne configure le tunnel IPv6-dans-IPv4. Notez que, grâce à l'autre tunnel, celui en GRE, la machine croit que `216.66.80.26` est directement joignable.

On a donc bien deux tunnels emboîtés :

```
% ip tunnel show
tun0: gre/ip  remote 41.204.93.101  local 172.16.0.76  ttl 255
he-ipv6: ipv6/ip  remote 216.66.80.26  local 41.204.94.222  ttl 255  6rd-prefix 2002::/16
```

À partir de là, IPv6 passe : on peut faire un `ping6 -n www.ietf.org`, on peut regarder le serveur Web du RIPE-NCC <http://www.ripe.net/> et voir en haut à droite qu'on utilise bien IPv6, etc. Là encore, si cela ne marche pas, les outils classiques, tcpdump, ping et traceroute sont là. Voici le trafic sur le câble physique Ethernet, vu par tcpdump :

```
12:11:24.217673 IP 41.204.94.222 > 216.66.80.26: \
          IP6 2001:470:1f09:1c8f:21c:23ff:fe00:6b7f > 2001:660:3003:2::4:20: \
              ICMP6, echo request, seq 3, length 64
12:11:24.364372 IP 41.204.93.101 > 172.16.1.2: GREv0, length 128: \
          IP 216.66.80.26 > 41.204.94.222: \
              IP6 2001:660:3003:2::4:20 > 2001:470:1f09:1c8f:21c:23ff:fe00:6b7f: \
                  ICMP6, echo reply, seq 3, length 64
```

On y voit `2001:470:1f09:1c8f:21c:23ff:fe00:6b7f` (la machine interne) pinguer `2001:660:3003:2::4:20` (un serveur quelque part sur l'Internet). On voit bien le routage asymétrique. La requête a été encapsulée une seule fois, dans le tunnel Hurricane Electric (de `41.204.94.222` vers `216.66.80.26`). La réponse, en revanche, a été encapsulée deux fois, dans le tunnel Hurricane Electric, puis dans le tunnel GRE. On note que tcpdump n'a pas de problème avec cette double encapsulation et arrive à décapsuler sans problème, affichant ce qui passe dans le tunnel. Si on regardait, toujours avec tcpdump, non pas sur l'interface physique `eth0` mais sur les interfaces virtuelles du tunnel (option `-i` de tcpdump), on verrait directement le trafic IPv6.

Mais, pour l'instant, seul le routeur IPv6, notre brave PC/Debian, peut profiter de l'Internet en IPv6. Les autres machines du réseau local n'y ont pas droit. Notre PC n'est pas encore un routeur. Pour cela, il faut d'abord activer le routage IPv6 en mettant les variables sysctl `net.ipv6.conf.all.forwarding` et `net.ipv6.conf.default.forwarding` à 1 (cela peut se faire avec la commande `sysctl` - option -w - ou bien en éditant `/etc/sysctl.conf` et en rechargeant les paramètres avec `sysctl -p`). Autrement, la machine jetterait sans remords les paquets IPv6 qui ne lui sont pas destinés.

Ensuite, il faut créer une route vers le réseau local, soit en dotant le routeur d'une adresse de ce réseau, soit simplement en ajoutant une route :

```
ip route add 2001:470:1f09:1c8f::/64 dev eth0
```

Enfin, il faut prévenir les machines du réseau local qu'un routeur IPv6 est là pour les servir. Cela se fait en envoyant des RA ("*Router Advertisment*", RFC 4862). J'ai installé le paquetage Debian `radvd` qui fournit ce service. La configuration est triviale :

```
# /etc/radvd.conf
interface eth0
{
   AdvSendAdvert on;
   prefix      2001:470:1f09:1c8f::/64 {};
};
```

Et c'est magique : toutes les machines du réseau local, par le biais de ces RA, reçoivent le préfixe du réseau et l'adresse du routeur. Désormais, tout le monde peut faire un `traceroute6 www.afnic.fr`.

La plaie traditionnelle des tunnels est qu'ils diminuent la MTU (à cause des quelques octets nécessaires pour l'en-tête du protocole d'encapsulation). Cela va donc souvent imposer de la fragmentation qui, en raison de l'incompétence de certains ingénieurs système qui bloquent l'ICMP, marche souvent mal sur l'Internet <https://www.bortzmeyer.org/mtu-et-mss-sont-dans-un-reseau.html>. Alors, avec deux tunnels emboîtés, cela risque d'être encore pire. Mais, si tout est correctement configuré (en gros, si on laisse ICMP passer sur tout le chemin), tout se passe bien. Testons depuis l'extérieur en demandant à ping des paquets plus gros que d'habitude <https://www.bortzmeyer.org/ping-taille-compte.html> pour qu'ils dépassent, avec les octets des en-têtes, la MTU :

```
% ping6 -s 1450 2001:470:1f09:1c8f:21d:92ff:fe7f:e0ab
PING 2001:470:1f09:1c8f:21d:92ff:fe7f:e0ab(2001:470:1f09:1c8f:21d:92ff:fe7f:e0ab) 1450 data bytes
From 2001:470:0:67::2 icmp_seq=1 Packet too big: mtu=1480
From 2001:470:0:67::2 icmp_seq=3 Packet too big: mtu=1456
1458 bytes from 2001:470:1f09:1c8f:21d:92ff:fe7f:e0ab: icmp_seq=5 ttl=59 time=138 ms
1458 bytes from 2001:470:1f09:1c8f:21d:92ff:fe7f:e0ab: icmp_seq=6 ttl=59 time=141 ms
```

Tout s'est bien passé. Les messages "*packet too big*" montrent que les paquets ICMP émis par le routeur d'entrée du tunnel (deux messages, un par tunnel) sont bien arrivés, ping en a tenu compte et tout fonctionne. (Pour les paquets émis depuis le réseau local, on pourrait s'épargner cela en configurant `radvd` pour n'indiquer qu'une MTU assez petite pour passer dans le tunnel.)

PS : ce réseau étant de nature expérimental, ne vous étonnez pas si vous testez les adresses et que cela « ne marche pas ».

Quelques lectures possibles :

— "*Le HOWTO sur GRE*" <http://lartc.org/lartc.html#LARTC.TUNNEL.GRE>,

— « "*Setting up GRE tunnel between cisco ios routers*" <https://dakeung.com/2009/12/24/setting-up-gre-tunnel-between-cisco-ios-routers/> » qui explique bien le côté IOS de la chose (et dont la seconde partie montre comment sécuriser le tunnel GRE avec IPsec).

Merci à Janvier Ngnoulaye pour sa patience et sa participation active et compétente.