RFC 6382 : Unique Per-Node Origin ASNs for Globally Anycasted Services

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 25 octobre 2011. Dernière mise à jour le 26 octobre 2011

Date de publication du RFC : Octobre 2011

https://www.bortzmeyer.org/6382.html

Ce court RFC propose un changement radical dans la gestion des annonces BGP par les services "*anycastés*". Actuellement, ils utilisent presque tous un seul numéro d'AS pour tous les nœuds d'un nuage "*anycast*". Notre RFC 6382 [1], au contraire, suggère d'utiliser un numéro d'AS par nœud.

Comme l'explique la section 2 de notre RFC, l'"*anycast*" (RFC 4786) est désormais une technique banale. Elle permet notamment d'augmenter sérieusement la résistance aux dDoS. Dans le monde du DNS, l'"*anycast*" est particulièrement répandu (par exemple, `.fr` n'a plus que deux serveurs "*unicasts*", tous les autres sont "*anycastés*").

L'"*anycast*" fonctionnant en injectant la même route, via BGP, depuis des points différents du réseau, on peut se demander s'il faut que tous ces points annonçent le préfixe depuis le même AS, ou bien si chacun doit utiliser un AS différent ? Aujourd'hui, la quasi-totalité des services "*anycastés*" utilisent un seul numéro d'AS pour tous leurs points de présence (POP). VeriSign (la boîte des auteurs du RFC) est un des rares acteurs à le faire, pour certains de leurs serveurs. Cela a notamment l'avantage de préserver une ressource rare : les numéros d'AS n'étaient codés que sur seize bits. Cela facilite également la configuraion des routeurs. Et, aujourd'hui, cela évite des ennuis avec les systèmes d'alarme BGP <https://www.bortzmeyer.org/alarmes-as.html>, qui pourraient couiner en voyant le même préfixe avoir plusieurs AS d'origine. Hurricane Electric fournit ainsi une liste des préfixes annoncés par plusieurs AS <http://bgp.he.net/report/multi-origin-routes>. Même chose chez Cymru <http://www.cymru.com/BGP/incon_asn_list.html>. Comme illustré par un exposé à NANOG en

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6382.txt

2001 <http://www.nanog.org/meetings/nanog23/abstracts.php?nm=nanog23&pt=OTc3Jm5hbm9nMjM=>, c'était plutôt considéré comme une erreur.

Mais cette politique a aussi des défauts : lorsqu'un routeur BGP voit arriver une annonce pour un préfixe "*anycasté*", il ne sait pas exactement de quel nœud elle vient. Cela peut compliquer le débogage des problèmes. Certains services ont un mécanisme pour identifier le serveur (RFC 5001 pour le DNS, ou bien le traditionnel quoique non normalisé `hostname.bind` dans la classe `CH`). Et, évidemment, on peut toujours utiliser traceroute pour trouver où se trouve telle instance "*anycast*". Essayons avec un serveur DNS "*anycasté*", `d.nic.fr`, depuis une machine située dans le Missouri :

```
% dig +nsid @d.nic.fr SOA fr.
...
; NSID: 64 6e 73 2e 6c 79 6e 32 2e 6e 69 63 2e 66 72  (d) (n) (s) (.) (l) (y) (n) (2) (.) (n) (i) (c) (.) (f
...
```

C'est donc `dns.lyn2.nic.fr` qui a répondu. Et on confirme (l'AFNIC utilise les Locodes pour nommer ses machines, donc LYN = Lyon) qu'on touche l'instance de Lyon (cette machine a surtout des instances en métropole et dans les DOM-TOM et aucune aux États-Unis) :

```
# tcptraceroute d.nic.fr 53
Selected device eth0, address 208.75.84.80, port 43778 for outgoing packets
Tracing the path to d.nic.fr (194.0.9.1) on TCP port 53 (domain), 30 hops max
 1  208.75.84.1  0.000 ms  0.000 ms  0.000 ms
 2  host123.datotel.com (208.75.82.123)  0.000 ms  0.000 ms  0.000 ms
 3  stl-c1-g1-15.datotel.com (208.82.151.29)  10.000 ms  0.000 ms  0.000 ms
 4  stl-e2-g0-2.datotel.com (208.82.151.13)  0.000 ms  0.000 ms  0.000 ms
 5  vlan100.car2.StLouis1.Level3.net (4.53.162.121)  0.000 ms  0.000 ms  0.000 ms
 6  ae-4-4.ebr2.Chicago1.Level3.net (4.69.132.190)  10.000 ms  9.999 ms  10.000 ms
 7  ae-5-5.ebr2.Chicago2.Level3.net (4.69.140.194)  0.000 ms  9.999 ms  9.999 ms
 8  ae-6-6.ebr2.Washington12.Level3.net (4.69.148.145)  10.000 ms  19.999 ms  19.999 ms
 9  ae-5-5.ebr2.Washington1.Level3.net (4.69.143.221)  19.999 ms  19.999 ms  29.998 ms
10  ae-44-44.ebr2.Paris1.Level3.net (4.69.137.61)  89.994 ms  99.994 ms  109.994 ms
11  ae-22-52.car2.Paris1.Level3.net (4.69.139.227)  109.994 ms  99.994 ms  99.994 ms
12  JAGUAR-NETW.car2.Paris1.Level3.net (212.73.207.162)  109.993 ms  99.995 ms  99.994 ms
13  dns.lyn2.afnic.cust.jaguar-network.net (78.153.224.126)  119.993 ms  119.993 ms  139.992 ms
14  d.nic.fr (194.0.9.1) [open]  109.994 ms  119.993 ms  109.993 ms
```

Les machines de l'AFNIC ayant un enregistrement DNS indiquant leur position physique (RFC 1876), on peut même être plus précis :

```
% dig LOC dns.lyn2.nic.fr
...
dns.lyn2.nic.fr. 172800 IN LOC 45 43 20.568 N 4 51 39.816 E 1.00m 1m 10m 10m
```

et on sait alors où est la machine.

Autre essai, avec un serveur de la racine du DNS, `L.root-servers.net`, largement "*anycasté*". Depuis un fournisseur en France :

```
% dig +nsid @l.root-servers.net SOA .
...
; NSID: 6c 79 73 30 31 2e 6c 2e 72 6f 6f 74 2d 73 65 72 76 65 72 73 2e 6f 72 67  (l) (y) (s) (0) (1) (.) (l)
-) (s) (e) (r) (v) (e) (r) (s) (.) (o) (r) (g)
```

On touche `lys01.l.root-servers.org`. Comme son opérateur, l'ICANN, utilise (comme beaucoup), les codes aéroport pour nommer les machines, on voit qu'elle est également à Lyon (LYS est l'aéroport de cette ville). Depuis une machine d'un autre FAI français, la même requête renvoie `lux01.l.root-servers.` ce qui situe le nœud "*anycast*" au Luxembourg. Et, en testant depuis la machine missourienne citée plus haut, on atteint `lax12.l.root-servers.org` soit Los Angeles.

Ces techniques sont toutefois imparfaites. Or, les services "*anycast*" ont des vulnérabilités paticulières. Par exemple, l'injection de routes pirates dans BGP par un méchant est plus difficile à détecter (cf. section 5). L'"*anycast*" a besoin d'outils de débogage puissants, pour venir à bout des problèmes de routage, volontaires ou involontaires, qui peuvent se manifester. Pire, on peut avoir des cas où les différentes instances d'un même nuage <`https://www.bortzmeyer.org/detournement-racine-pekin.html`> "*anycast*" ne répondent pas exactement la même chose, et il est dans ce cas crucial de pouvoir identifier sans aucune ambiguïté celles qui sont différentes.

Avant la recommandation officielle, un petit détour de terminologie (section 1). Parmi les termes importants (lire le RFC 4786 pour plus de détails) :

— **Bassin d'attraction** ("*catchment*"), terme emprunté à l'hydrographie (la zone dont toutes les eaux finissent dans une rivière donnée), désigne la partie de l'Internet qui envoie ses paquets à un nœud (une instance "*anycast*") donné,
— Nœud local, instance d'un service "*anycast*" qui ne publie son préfixe que dans une partie de l'Internet (typiquement, uniquement les participants à un point d'échange donné, en utilisant la communauté BGP `no-export`). La recommandation de ce RFC s'applique à tous les nœuds, locaux ou globaux,
— Nœud global, instance qui annonce son préfixe à l'Internet entier.

Venons-en maintenant à la nouveauté de ce RFC. Il tient en une phrase (section 3), « **Il faudrait utiliser un numéro d'AS différent par nœud** ». Le but est de fournir un mécanisme discriminant les annonces. Si on a deux nœuds, un en Chine (AS 65536) et un en Europe (AS 65551), et qu'on voit le préfixe "*anycast*" `192.0.2.64/26` annoncé depuis l'AS 65536, on sait qu'on s'adressera à l'instance chinoise. On peut même filtrer sur l'AS d'origine pour éviter cela <`https://www.bortzmeyer.org/detournement-racine-pekin.html`>. L'utilisation de numéros d'AS différents permettra de choisir sa politique de routage.

Est-ce sans inconvénient ? Actuellement, le principal problème risque d'être les systèmes d'alarme <`https://www.bortzmeyer.org/alarmes-as.html`> qui s'inquiéteraient de ces différentes origines. Ainsi, BGPmon <`http://bgpmon.net/`>, par défaut, considère qu'une annonce depuis un autre AS que celui indiqué comme origine, est une attaque possible ("*possible hijack*") et il alarme. Toutefois, le même BGPmon permet d'indiquer plusieurs AS d'origine supplémentaire, ce qui lui permet de gérer la nouvelle politique. (Une société comme PCH a environ soixante localisations physiques dans le monde, Netnod <`http://www.netnod.se/`> en a cent : les enregistrer toutes comme origine possible, auprès d'un système d'alarme BGP, pourrait être fastidieux. À noter que la fonction "*auto-detect*" de BGPmon simplifie cela : comme l'explique l'auteur « "*Just click on the prefix to edit it, then click on the little green icon next to the 'Additional Origin AS' section. It will then show a popup with all additional origin ASn's we have in our database. You can then copy paste that line into the 'Additional Origin AS' field.*" ». Un exemple est donné par un serveur de `.com`, m.gtld-servers.net <`http://www.bgpmon.net/autodetect_origin.php?prefix=192.55.83.0&length=24&originas=36618`>.)

Autre inconvénient possible : la consommation excessive de numéros d'AS. Toutefois, depuis le RFC 4893, ceux-ci peuvent désormais être codés sur 32 bits et le risque d'épuisement disparait.

Après cette nouvelle recommandation de consommer un numéro d'AS par site, la section 4 du RFC rassemble divers conseils aux gérants de services "*anycast*" :

— Publier des informations sur la localisation physique de la machine (la méthode n'est pas précisée mais on peut penser aux enregistrements LOC du RFC 1876, peut-être en les attachant au nom obtenu par la requête NSID),
— Publier dans les IRR les informations sur les AS auxquels le nœud "*anycast*" s'interconnecte (par exemple en RPSL, RFC 4012).

Et la section 6 contient l'examen de divers points pratiques liés au déploiement de cette nouvelle politique. Les opérateurs de services "*anycast*" critiques ont été largement consultés avant la publication du RFC, ce qui ne veut pas dire qu'ils déploieront cette recommandation tout de suite (aucun n'a annoncé de plan précis et l'ISC a au contraire annoncé qu'ils ne le feraient pas <https://www.isc.org/community/blog/201109/origin-asn-anycasted-services>, voir plus loin leur analyse). En gros, la nouvelle politique fera davantage de travail au début (obtenir les numéros d'AS - ce qui nécessitera probablement un changement dans la politique des RIR, ajouter une colonne pour le numéro d'AS dans la base de données des instances "*anycast*", penser à indiquer le bon numéro d'AS lorsqu'on fait une demande de "*peering*", changer les centaines de "*peerings*" existants, etc) mais simplifiera la surveillance du service, en permettant de trouver plus facilement l'origine d'une annonce.

Et pour finir, un exemple de ce que donne un excellent outil d'analyse existant, le RIS <http://www.ris.ripe.net/>, avec le serveur de .com déjà cité (annoncé par trois AS) : <http://www.ris.ripe.net/dashboard/192.55.83.0/24>.

Pour avoir un autre point de vue, l'ISC a expliqué le fonctionnement de l'anycast chez eux <http://www.isc.org/community/blog/201008/f-root-routing-how-does-it-work-0>, une explication détaillée de la supériorité de leur système <http://www.mail-archive.com/grow@ietf.org/msg00886.html>, ainsi que leur liste de peerings <http://www.isc.org/community/peering>.

Une autre question n'est pas couverte dans le RFC mais mérite une mention (merci à Olivier Benghozi et Guillaume Barrot pour leurs explications) : pourquoi n'avoir pas utilisé plutôt les communautés BGP (RFC 1997), des étiquettes qu'on peut attacher aux annonces et qui sont transitives ? La raison principale est qu'elles sont fréquemment effacées en entrée des AS (sans compter les systèmes qui, par défaut, ne les transmettent pas du tout comme IOS, cf. un article sur leur utilisation <http://blog.ioshints.info/2009/05/bgp-basics-bgp-communities-propagation.html>). Même problème avec d'autres attributs facultatifs de BGP comme AGGREGATOR (qu'on aurait pu, en le détournant un peu, utiliser à ce but).

Merci à Jean-Philippe Pick pour sa relecture et pour les informations.