RFC 4786 : Operation of Anycast Services

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 26 décembre 2006. Dernière mise à jour le 28 décembre 2006

Date de publication du RFC : Décembre 2006

https://www.bortzmeyer.org/4786.html

L'"*anycast*" : un terme complètement inconnu il y a trois ans, qui est maintenant très souvent prononcé. Cette technique de distribution de serveurs a largement fait ses preuves et est maintenu reconnu comme « bonne pratique ».

Notons que le terme d'"*anycast*" est officiellement traduit (Journal Officiel de la République Française <http://www.journal-officiel.gouv.fr/>, 28 décembre 2006, Commissions générale de terminologie et de néologie, « Vocabulaire des télécommunications ») par « envoi à la cantonade ».

Le premier RFC a avoir parlé de l'"*anycast*" est le RFC 1546 [1], il y a plus de dix ans. Mais la technique n'a vraiment décollé qu'à partir de la grande attaque d'octobre 2002 <http://www.isc.org/f-root-denial-of-service-21-oct-2002> qui a mis en évidence la vulnérabilité des indispensables serveurs de la racine du DNS. Quelques mois après, les premiers serveurs racine "*anycastés*" voyaient le jour, nouvel hommage à la réactivité de l'Internet. La première documentation moderne était l'excellente note technique de l'ISC : "*Hierarchical Anycast for Global Service Distribution*" <http://www.isc.org/pubs/tn/isc-tn-2003-1.html> de Joe Abley, un des auteurs de notre RFC. Ce succès (l'attaque d'octobre 2002 ne s'est jamais reproduite) est largement dû aux efforts de l'ISC.

Le but principal de l'"*anycast*" est donc la sécurité, notamment la résistance aux attaques DoS. Cette résistance accrue est due au plus grand nombre de serveurs qui peut être déployé, grâce à l'"*anycast*", et à la concentration des attaques dans un lieu proche des attaquants, ce qui permet de les détecter plus facilement.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc1546.txt

En quoi consiste l'"*anycast*" ? Le principe est de disposer plusieurs serveurs, tous aptes à rendre le même service, et de permettre au client de choisir automatiquement le « meilleur ». Contrairement à la technique des miroirs, l'"*anycast*" se fait sans intervention du client. Contrairement aux systèmes de répartition de charge, la requête du client ne passe par aucune machine qui déciderait du serveur choisi. Le service de loin le plus "*anycasté*" est le DNS et la méthode la plus courante repose sur le protocole de routage BGP : le même préfixe (par exemple `192.5.5.0/24` pour `F.root-servers.net`) est annoncé par les routeurs de chaque site où se trouve une **instance** de `F.root-servers.net`. Le traditionnel algorithme de sélection des routes par BGP choisira l'instance "*anycast*" qui servira la requête. En général, les dites instances sont situées sur des points d'échange et l'instance "*anycast*" sert les opérateurs connectés à ce point.

Notre RFC n'explique pas comment on construit un service "*anycast*" (on trouve des informations utiles dans "*A Software Approach to Distributing Requests for DNS Service using GNU Zebra, ISC BIND 9 and FreeBSD*" <http://www.isc.org/pubs/tn/isc-tn-2004-1.html> ou bien sur le serveur du projet AS112 <http://www.as112.net/>). Il explique les bonnes pratiques, pour être sûr que les avantages de l'"*anycast*" l'emportent sur ses inconvénients.

Notre RFC commence donc par étudier les protocoles qui bénéficient le plus de l'"*anycast*" : le DNS, étant typiquement sans état (chaque requête est indépendante des autres) est le plus adapté et ce n'est pas un hasard s'il est le plus "*anycasté*" aujourd'hui. Les protocoles où les sessions peuvent être longues (SSH, par exemple) sont difficiles à "*anycaster*" : un changement des tables de routage et les paquets arrivent à une autre instance, qui ne sait pas comment reprendre la session.

Ensuite, le RFC examine des questions comme le placement des nœuds (instances) "*anycast*".

Un gros morceau commence ensuite, dans les sections 4.3 et 4.4 : le routage. L'"*anycast*" peut fonctionner avec un protocole de routage externe comme BGP mais aussi avec un protocole de routage interne comme OSPF. Dans les deux cas, il faut coupler étroitement le routeur et le serveur, de façon à ce qu'un arrêt du serveur, entraine l'arrêt de l'annonce de la route. Le RFC détaille ensuite le cas de certains mécanismes comme PPLB <http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s21/pplb.htm> ou comme RPF (décrit dans le RFC 3704) qui, dans certains cas très rares sur Internet, peuvent interférer avec l'"*anycast*".

Le RFC détaille ensuite, notamment dans sa section 5, des considérations plus pratiques d'administration système : surveillance d'un service "*anycast*" (il faut surveiller chaque instance individuellement), débogage ("*anycast*" est conçu pour que les instances soient indistinguables par les clients, ce qui peut compliquer la recherche sur un problème), etc.

La sécurité étant une des principales motivations de l'"*anycast*", on ne s'étonnera pas que le RFC se termine par une longue section sur ce sujet, qui insiste entre autres sur les risques nouveaux liés à l'"*anycast*" ; par exemple, une annonce BGP pirate (RFC 4272) sera moins facilement détectée qu'avec un service traditionnel.

Notons pour terminer que le site Web <http://www.root-servers.org/> présente une liste des serveurs racines actuellement "*anycastés*" et la localisation de leurs instances.