RFC 7094 : Architectural Considerations of IP Anycast

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 17 janvier 2014

Date de publication du RFC : Janvier 2014

https://www.bortzmeyer.org/7094.html

La technologie de l'anycast est maintenant largement déployée dans l'Internet. C'est l'occasion de faire un point architectural : quelles sont les conséquences de l'anycast ? Telle est la question de ce RFC de l'IAB.

Qu'est-ce que l'anycast ? C'est simplement le fait qu'une adresse IP, celle à laquelle répond un service offert sur l'Internet, soit configurée en plusieurs endroits (cf. RFC 4786 [1]). Cette technique est surtout connue par son utilisation massive dans le DNS, notamment pour les serveurs racine. Fin 2007, il y avait déjà dix des treize serveurs racine qui utilisaient l'anycast (cf. les minutes de la réunion 29 du RSSAC <https://www.icann.org/en/groups/rssac/meetings/rssac-29-en.pdf>). Sans compter la quasi-totalité des gros TLD qui reposent, eux aussi, sur cette technique pour tout ou partie de leurs serveurs. À part le DNS, l'utilisation la plus importante de l'anycast est sans doute pour les serveurs NTP (RFC 5905).

L'adresse IP perd donc sa caractéristique « est unique dans tout l'Internet ». La même adresse IP est configurée en plusieurs endroits, et un paquet IP à destination de cette adresse IP de service arrivera dans un seul de ces endroits, le « plus proche » (en termes de métriques de routage, pas forcément en terme de distance géographique ou de RTT).

À noter que rien ne distingue, syntaxiquement, une adresse anycast d'une adresse habituelle. D'ailleurs, déterminer si un service donné est « anycasté » n'est pas toujours facile. (Quelques traceroutes depuis divers points, ainsi qu'une interrogation directe du serveur, par exemple avec le RFC 5001 pour le DNS, peuvent répondre à cette question, mais on n'a pas toujours une certitude.)

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc4786.txt

Le RFC 4786 discute en détail différentes façons de faire de l'anycast. Notre nouveau RFC 7094 ajoute une nouvelle distinction : « "*off-link anycast*" » et « "*on-link anycast*" ». Le premier mode est celui de l'anycast actuel, largement déployé comme on l'a vu. Plusieurs sites annoncent la même route et le « plus proche » est choisi. C'est ce mode qui est discuté dans le RFC 4786.

Le second mode, "*on-link anycast*", n'est pas réellement déployé. Il désigne un système où le protocole de résolution d'adresses sur un réseau local peut gérer, sans protocole de routage, l'envoi à une machine parmi celles qui répondent à une adresse IP donnée. Dans le monde IP, il n'est normalisé que pour IPv6 (RFC 4861, notamment la section 7.2.7).

Lorqu'on utilise le premier mode, le "*off-link anycast*", quel AS d'origine doit-on mettre dans l'annonce BGP ? La plupart des sites anycast utilisent le même AS pour chaque site mais le RFC 6382 documente une méthode où un AS différent est utilisé par site (méthode « multi-origines »).

De quand date l'anycast ? La section 2.1 se penche sur l'histoire de ce concept. La première spécification était le RFC 1546 en 1993, ce qui ne nous rajeunit pas. À l'époque, c'était purement théorique mais le RFC 1546 liste bien toutes les questions. La première utilisation documentée était l'année suivante, pour de la distribution de vidéo, et est notée dans l'IMR 9401 <ftp://ftp.rfc-editor.org/in-notes/museum/imr/imr9401.txt>. Il s'agissait de trouver le serveur de vidéos « le plus proche ». La rumeur de l'Internet dit que des FAI ont utilisé l'anycast pour leurs résolveurs DNS à partir de cette époque mais il n'y a pas de traces écrites. En 1997, l'IAB s'en mêle pour la première fois, avec quelques phrases dans le RFC 2101.

L'anycast a également été mentionné lors de l'atelier sur le routage organisé par l'IAB l'année suivante et documenté dans le RFC 2902. Cette technique est encore présentée comme « à étudier » (« "*We need to describe the advantages and disadvantages of anycast*" »). Donc, en 1999, à la 46ème réunion de l'IETF, une BoF sur l'anycast a eu lieu. Les deux principales conclusions étaient que l'usage de l'anycast pour TCP n'était pas forcément une bonne idée, mais que par contre l'application qui serait la plus suceptible d'utiliser l'anycast avec profit était le DNS.

Par la suite, IPv6 est arrivé et a spécifié des adresses anycast dans le RFC 2526 et dans le RFC 3775. Il était aussi prévu d'avoir des relais anycastés pour la coexistence IPv4-IPv6 (RFC 2893) mais cela a été supprimé par la suite, avant que 6to4 ne réintroduise l'idée dans le RFC 3068. L'expérience de 6to4 a d'ailleurs bien servi à mesurer les conséquences de l'utilisation de l'anycast « en vrai » et cela a été documenté dans le RFC 3964.

Mais c'est le DNS qui a popularisé l'anycast, d'abord avec l'AS 112 (décrit dans le RFC 7534) puis avec la racine. En 2002, le RFC 3258 décrit cette utilisation. Il est amusant de constater que ce RFC n'utilise pas le terme anycast, considéré à l'époque comme devant désigner uniquement les techniques reposant sur une infrastructure spécifique (comme le multicast). Le RFC 3258 note aussi qu'aucun changement des mécanismes de routage n'est nécessaire. Et que l'utilisation d'UDP par le DNS suffisait à résoudre le problème du changement des routes en cours de session : le DNS étant requête/réponse, sur un protocole (UDP) sans état, il n'y avait pas de risque de parler à deux instances différentes pendant une session. Le RFC 3258 recommandait d'arrêter le serveur DNS, plutôt que de couper le routage, en cas de panne, arguant du fait que les clients DNS savent se débrouiller avec un serveur en panne. Couper le routage a en effet l'inconvénient de propager des mises à jour BGP dans toute la DFZ, ce qui peut mener à leur rejet (RFC 2439). Cet argument ne vaut que pour les serveurs faisant autorité (se débrouiller avec un résolveur en panne est bien plus lent). Aujourd'hui, on conseille plutôt le contraire, notamment lorsqu'on a des SLA par serveur DNS (ce qui est absurde, puisque le DNS se débrouille très bien avec un serveur en panne, mais cela se voit dans certains contrats). La section 4.5 de notre RFC revient plus en détail sur ce conseil.

Sur la base notamment du déploiement de l'anycast dans le DNS de 2002 à 2006 a été écrit le RFC 4786, sur les aspects pratiques de l'anycast. Il insiste sur la nécessité que le routage soit « stable », c'est-à-dire dure plus longtemps que la transaction typique (ce qui est facile à réaliser avec les protocoles requête/réponse comme le DNS). Il note aussi que l'anycast peut compliquer la gestion des réseaux, en ajoutant de nouvelles causes de perturbation.

Il y a d'autres cas où le vocabulaire a changé. Par exemple, IPv6 venait dès le début avec quelque chose nommé anycast mais qui était uniquement du "*on-link anycast*", pas le "*off-link anycast*" qui est massivement utilisé par le DNS. Cela impliquait des restrictions sur l'usage (interdiction de mettre une adresse anycast en source d'un paquet, cf. section 2.5 du RFC 1884) qui ont été supprimées en 2006 par le RFC 4291. L'"*on-link anycast*" d'IPv6 existe toujours (section 7.2.7 du RFC 4861) mais semble très peu employé.

La section 3 de notre RFC pose ensuite les principes d'architecture à suivre pour l'anycast. Elle est assez restrictive, limitant l'anycast à quelques protocoles simples comme le DNS. D'abord, le modèle en couches est considéré comme essentiel par sa modularité. Cela veut dire qu'une application (couche 7) ne devrait pas avoir à être modifiée pour s'adapter au système de routage (couche 3). L'application ne devrait pas non plus être concernée par la stabilité des routes, comme elle n'est pas concernée par le taux de perte de paquets, qui est en général traité dans une couche plus basse. Avec les protocoles de transport à états, comme TCP, un changement de routage qui fait arriver à une autre instance et c'est la fin de tout, puisque cette instance n'a pas l'état associé : elle enverra purement et simplement un `RST`, mettant fin à la connexion. À noter que c'est un des cas où quelque chose peut très bien marcher dans le laboratoire et pas dans le vrai Internet (où les changements de routes sont un événement relativement fréquent).

Donc, si une adresse de destination d'un paquet est anycast, différents paquets à destination de cette adresse pourront être envoyés à des machines distinctes. Ce n'est pas un problème si on respecte les principes posés par ce RFC : une requête qui tient dans un seul paquet, un transport sans état (comme UDP), pas d'état côté serveur entre deux requêtes, et des requêtes idempotentes. Le DNS obéit à tous ces principes.

Et les adresses anycast en source et pas seulement en destination ? Si on fait cela, deux réponses au même paquet pourront arriver à deux machines différentes. Cela marchera quand même s'il n'y a pas de réponses (cas du DNS : le demandeur ne répond pas à la réponse, venant d'un serveur anycast). Ou si elles sont distribuées à la bonne instance par un autre mécanisme, spécifique à l'application. Mais il y a un autre piège avec une adresse anycast en source : si on met en œuvre le filtrage RPF (RFC 4778), le paquet entrant sera peut-être refusé puisque rien ne dit que le paquet d'une instance donnée arrivera par l'interface par lequel une réponse serait sortie. La section 4.4.5 du RFC 4786 donne quelques conseils à ce sujet.

Une solution au problème du changement de routage, amenant à une autre instance, serait de n'utiliser l'anycast que pour la découverte du serveur, pas pour la communication avec ce serveur (section 3.4). Cela ne convient pas au DNS, puisque cela augmente d'un aller-retour le temps total d'interaction mais ce n'est pas grave, le DNS ne craint pas les changements de route. En revanche, pour un protocole utilisant TCP, on pourrait envoyer un paquet UDP à l'adresse anycast de découverte, recevoir en réponse l'adresse unicast du serveur et interagir ensuite en TCP avec cette adresse unicast. Ainsi, on serait à l'abri des changements de route ultérieurs, tout en gardant les avantages de l'anycast, notamment la sélection du serveur le plus proche. Je ne connais pas encore de système qui utilise cette approche, pourtant prometteuse.

Enfin, la section 4 du RFC analyse certains points précis de l'anycast. Par exemple, son passage à l'échelle pour le système de routage. Chaque serveur anycast public sur l'Internet va nécessiter une

route dans la DFZ et va donc charger tous les routeurs de la DFZ. Il est donc préférable que l'usage de l'anycast reste limité à quelques services (comme c'est le cas actuellement). Certes, on peut mettre plusieurs serveurs derrière un même préfixe anycast (et donc une seule route) mais cela imposerait qu'à **chaque** site, on trouve la totalité des serveurs.

On a vu qu'on pouvait avoir le beurre (l'anycast) et l'argent du beurre (la résistance au changement de routes) en commençant les connexions TCP par un échange UDP pour trouver l'adresse du serveur le plus proche. Cela se paie par un aller-retour supplémentaire (celui en UDP). Il existe une façon astucieuse d'éviter ce coût mais elle requiert une modification de TCP. L'idée est la suivante : le premier paquet TCP, le paquet `SYN`, est envoyé à l'adresse anycast. La réponse, `SYN-ACK` est émise depuis une adresse source unicast. Le reste de la communication (fin de la triple poignée de mains avec un `ACK` puis échange de données) se ferait avec l'adresse unicast du serveur, à l'abri des fluctuations du routage. Les mises en œuvre actuelles de TCP rejetteraient le deuxième paquet, le `SYN-ACK` car l'adresse source ne correspond pas. Mais, si elles étaient modifiées pour accepter n'importe quelle adresse source en réponse à un `SYN` envoyé à une adresse anycast, cela marcherait (le paquet entrant serait envoyé au processus qui a une prise réseau avec les mêmes ports et un numéro de séquence TCP qui convient.) Modifier TCP est peut-être trop risqué mais la même idée pourrait être appliquée à des protocoles de transport plus récents comme SCTP.

La section 4.4 se penche sur les questions de sécurité. On utilise souvent l'anycast avant tout pour limiter les effets des dDoS, en ayant plusieurs sites pour amortir l'attaque, avec en outre un contingentement géographique (chaque zombie de l'attaquant ne peut taper que sur un seul site). On peut même s'en servir pour faire couler le trafic non désiré au fond de l'évier, dans l'esprit du RFC 3882. Cette méthode a bien marché dans le cas de grosses erreurs de configuration comme la bavure Netgear/NTP documentée dans le RFC 4085.

On a parlé plus haut du conseil de ce RFC de ne **pas** retirer une route en cas de panne mais de laisser le trafic arriver au serveur en panne. Ce conseil est contesté mais, en cas d'attaque, il n'y a guère de doute qu'il faut laisser la route en place, sinon l'attaque ira simplement vers une autre instance, probablement plus lointaine, en gênant davantage de monde au passage. Une exception est celle où on retire la route pour aider à trouver la vraie source d'une attaque <`https://www.bortzmeyer.org/identifier-spoofer.html`>.

Autre question de sécurité, le risque d'un faux serveur. Ce n'est pas un risque purement théorique, loin de là. Ainsi, à la Journée du Conseil Scientifique de l'AFNIC <`http://www.afnic.fr/fr/l-afnic-en-bref/actualites/actualites-generales/6171/show/succes-pour-la-journee-du-conseil-scientifi`
`html`> du 4 juillet 2012, Daniel Karrenberg avait expliqué que le RIPE-NCC avait identifié plusieurs copies pirates du serveur DNS racine `K.root-servers.net`. Bien sûr, annoncer un préfixe qui n'est pas à vous et attirer ainsi le trafic légitime vers le pirate a toujours été possible en BGP. Mais l'anycast rend cette attaque plus facile : si on voit une annonce d'un préfixe d'un ministère français par un opérateur biélorusse, on s'étonnera et on cherchera. Avec l'anycast, il est tout à fait normal de voir des annonces BGP (ou d'un protocole de routage interne) du même préfixe jaillir d'un peu partout, même d'AS différents. Il est donc très recommandé d'avoir un mécanisme d'authentification, permettant de s'assurer qu'on parle à une instance anycast légitime, et pas à un pirate. (Dans le cas de DNS, c'est DNSSEC qui assure ce rôle. Comme DNSSEC protège les données, et pas juste le canal, si la zone DNS est signée, peu importe que le résultat soit obtenu via un serveur légitime ou pas.)

D'autres attaques que l'envoi de réponses mensongères sont possibles en faisant des annonces de routage d'un préfixe utilisé en anycast. Par exemple, on peut empêcher les paquets d'atteindre les serveurs légitimes, leur masquant ainsi le trafic. La section 6.3 du RFC 4786 et l'article de Fan, X., Heidemann, J., et R. Govindan, « "*Evaluating Anycast in the Domain Name System*" <`http://www.isi.edu/~johnh/PAPERS/Fan13a/index.html`> » fournissent davantage de détails sur ces attaques spécifiques à l'anycast.