RFC 6586 : Experiences from an IPv6-Only Network

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 22 avril 2012

Date de publication du RFC : Avril 2012

https://www.bortzmeyer.org/6586.html

La plupart des RFC décrivent, de façon normative, un protocole ou un format qui sera ensuite déployé sur l'Internet. Mais ce document est différent : c'est un récit d'expérience, celle de deux courageux chercheurs qui, au péril de leur usage de Facebook et dans l'intérêt de la science, ont coupé IPv4 et n'ont accédé à l'Internet que via IPv6 pendant la durée de l'expérience. Les machines de l'expérience n'avaient plus du tout d'IPv4 et seule une fragile passerelle NAT64 les reliaient aux territoires barbares où régnait encore IPv4. Qu'est-ce qui marche dans ce cas ? Qu'est-ce qui ne marche pas ? Qu'est-ce qu'on peut améliorer ?

Deux réseaux ont été utilisés, un professionnel et un à la maison. Tous les deux étaient en double-pile (IPv4 **et** IPv6, cf. RFC 4213 [1]) depuis longtemps. Tous les deux avaient une connectivité IPv6 stable et correcte depuis des années. Il était donc temps de sortir de la « zone de confort » et d'essayer le grand large. Mais la majorité de l'Internet reste accessible en IPv4 seulement. C'est là qu'intervient NAT64 (RFC 6144), qui permet aux machines purement IPv6 d'accèder à des sites Web attardés en v4.

Ce n'est pas juste une expérience de démonstration pour montrer que les Vrais Hommes peuvent se passer du vieux protocole des mangeurs de yaourt. L'épuisement des adresses IPv4 <https://www.bortzmeyer.org/epuisement-adresses-ipv4.html> fait que de plus en plus de FAI envisagent de déployer des réseaux uniquement IPv6, puisqu'ils n'ont plus la possibilité d'obtenir des adresses. Tester de tels réseaux en vrai est donc nécessaire. La conclusion est d'ailleurs que cela ne marche pas trop mal mais qu'il reste un certain nombre de petits problèmes qu'il serait bon de régler.

La section 3 du RFC explique l'environnement de l'expérience. Les deux réseaux (un à la maison et un au bureau) étaient un mélange classique de PC, d'appareils photos numériques, de gadgets électroniques divers et de petits routeurs du commerce, avec divers systèmes (Mac OS, Windows, Linux...). Les usages

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc4213.txt

étaient également classiques, courrier, SSH, VoIP, jeu (pas dans le réseau de bureau, bien sûr), messagerie instantanée, un peu de domotique et bien sûr du Web. Il s'agissait de petits réseaux avec une dizaine d'utilisateurs au maximum.

Comme certains des utilisateurs des deux réseaux, des petits bras, avaient choisi de ne pas participer à l'expérience, il a fallu créer un réseau séparé pour celle-ci, avec un VLAN propre. Il n'y avait évidemment pas de serveur DHCP v4 sur ce réseau seulement des RA ("*Router Advertisement*") v6.

On l'a vu, l'essentiel du Web aujourd'hui n'est accessible qu'en IPv4. Il a donc fallu mettre en place une passerelle NAT64 (RFC 6146). Elle incluait le serveur DNS64 (RFC 6147). L'installation et la configuration de cette passerelle étaient très simples pour un administrateur réseaux.

Le serveur DNS cité plus haut était publié via les RA (RFC 6106). (À noter que Windows écrit aussi à des serveurs DNS bien connus, `fec0:0:0:ffff::1`, `fec0:0:0:ffff::2`, et `fec0:0:0:ffff::`.) Trouver le serveur DNS dans un environnement IPv6 pur n'est pas trivial. Dans les environnements mixtes, il est fréquent que le résolveur DNS à utiliser par les clients ne soit publié qu'en v4, les clients demandant ensuite aussi bien les enregistrements A que les AAAA en IPv4. Ici, cette méthode n'était pas possible et les auteurs du RFC ont également testé avec DHCP v6 (RFC 8415). Un des charmes d'IPv6 est en effet qu'il existe deux façons de découvrir automatiquement le résolveur DNS local, via les RA ou bien via DHCP.

Le serveur DNS64 et la passerelle NAT64 fonctionnaient de la manière habituelle (RFC 6144) : si un client demandait un AAAA (adresse IPv6 dans le DNS) inexistant, le serveur DNS64 en synthétisait un, les paquets à destination de cette adresse étaient ensuite interceptés par la passerelle et NATés en IPv4 (la passerelle avait, elle, une adresse v4 publique). Les destinations ayant des AAAA étaient traitées normalement : la passerelle servait alors de simple routeur v6, le cas idéal d'une connectivité v6 complète.

Les résultats de l'expérience occupent le reste du RFC. La section 4 fournit une synthèse et la section 5 examine un par un les choses qui ont plus ou moins bien marché. Principale conclusion ; **ça marche**. Un des auteurs du RFC travaille dans un environnement IPv6 pur depuis un an et demi et est toujours vivant et en bonne santé. Certains points marchent particulièrement bien (aucun problème avec le Web). Sur un téléphone Symbian, **toutes** les applications ont marché (sur un Android, toutes les applications de base marchaient, mais après un bricolage pour permettre à la machine de trouver son résolveur DNS).

Les problèmes rencontrés se subdivisent en plusieurs catégories :

— Les bogues pures et simples. Un logiciel gère théoriquement IPv6 mais, en pratique, on trouve des problèmes qui n'existent pas en v4. Ces bogues sont relativement fréquentes car trop peu de tests sont faits avec IPv6.
— Pas d'IPv6 du tout, même en théorie. Cela arrive à des programmes antédiluviens (le RFC cite dict, un client RFC 2229) mais aussi, et là c'est inexcusable, à des programmes plus récents comme le logiciel privateur Skype, ou comme beaucoup de jeux.
— Des problèmes liés au protocole. Dès que le protocole transmet des adresses IP, les ennuis commencent. La machine locale utilise IPv6, transmet l'adresse au pair v4 avec qui elle communique grâce au NAT64, pair qui n'arrive pas à répondre. Un exemple est fourni par certaines applications Web qui renvoient une page avec des URL utilisant des adresses IPv4 littérales (comme `http://192.0.2.80/`). Ce problème est rare mais très agaçant lorsqu'il se produit.
— Problèmes situés quelque part dans la couche 3, le plus fréquent étant les problèmes de MTU causés par un pare-feu mal configuré.

Bref, aucun de ces problèmes n'était un problème fondamental d'IPv6 ou de NAT64, nécessitant de revisiter le protocole. C'était « uniquement » des questions de développement logiciel.

Maintenant, avec la section 5, voyons ces problèmes cas par cas. D'abord, ceux situés dans les systèmes d'exploitation. Par exemple, Linux ne jette pas l'information acquise par les RA lorsque la connectivité réseau change, il faut faire explicitement un cycle suspension/reprise. Ensuite, le démon rdnssd, censé écouter les annonces de résolveurs DNS faites en RA, n'était pas intégré par défaut sur Ubuntu et, de toute façon, ne semblait pas très fiable. Comme pas mal de problèmes rencontrés lors de cette expérience, il s'est résolu en cours de route, avec une nouvelle version du système. Mais tout n'est pas parfait, notamment le NetworkManager qui s'obstine à sélectionner un réseau sans-fil IPv4 qui n'a pas de connectivité externe, plutôt que le réseau IPv6 pur.

Les autres systèmes ont aussi ce genre de problèmes. Ainsi, avec Mac OS X, il faut dire explicitement au système qu'IPv4 est facultatif et qu'un réseau sans IPv4 ne doit pas être considéré comme cassé et à ignorer. Et passer d'un réseau v4 à v6 nécessite des manipulations manuelles.

Pour Windows 7, le problème était avec les résolveurs DNS : Windows les affichait mais ne les utilisait pas sans action manuelle.

Android a un problème analogue. Il peut faire de l'IPv6 mais il ne sait pas sélectionner les résolveurs DNS en IPv6. Par défaut, il lui faut donc un résolveur DNS v4. Heureusement, il s'agit de logiciel libre et les auteurs ont pu résoudre ce problème eux-même <http://www.arkko.com/ddd/> avec le nouveau logiciel DDD.

En conclusion, le RFC note que tous ces systèmes ont IPv6 à leur catalogue depuis des années, parfois de nombreuses années, mais tous ces problèmes donnent à penser qu'ils n'ont jamais été sérieusement testés.

La situation des langages de programmation et des API semble meilleure, par exemple Perl qui était un des derniers grands langages à ne pas gérer complètement IPv6 par défaut semble désormais correct.

C'est moins satisfaisant pour la messagerie instantanée et la voix sur IP. Les cris les plus perçants des utilisateurs impliqués dans l'expérience avaient été provoqués par Skype, qui ne marche pas du tout en IPv6 (il a fallu utiliser un relais SSH vers une machine distante). La légende comme quoi Skype marcherait toujours du premier coup en prend donc un... coup.

Parmi les autres solutions testées, celles passant par le Web marchaient (Gmail ou Facebook, par exemple), celles fondées sur XMPP également mais les solutions commerciales fermées comme MSN, WebEx ou AOL échouent toutes. Ces solutions sont mises en œuvre dans des logiciels privateurs (on ne peut donc pas examiner et corriger le source) mais l'examen du trafic réseau montre qu'il passe des adresses IPv4 entre machines, ce qui est incompatible avec NAT64.

Et pour les applications stratégiques essentielles, je veux dire les jeux ? Ceux utilisant le Web n'ont pas de problème, pour les autres, c'est l'échec **complet**. En outre, aucun des ces logiciels ne produit de messages de diagnostic utilisables et le déboguage est donc très difficile. Au moment des premiers tests, ni Battlefield, ni Age of Empires, ni Crysis ne fonctionnaient sur le nouveau réseau. Depuis, World of Warcraft est devenu le premier jeu majeur qui marchait en IPv6. Les jeux dont le source a été libéré (comme Quake) ont également acquis cette capacité. Les autres feraient bien d'utiliser une API réseau plus moderne... (Cf. RFC 4038.)

Pendant qu'on en est au divertissement, que devient la musique en ligne ? La plupart des boutiques reposent sur le Web et marchent donc, sauf Spotify qui réussit l'exploit d'être un des rares services accessibles via le Web qui ne fonctionne pas en IPv6.

Moins bonne est la situation des "*appliances*" comme les webcams. La plupart ne parlent pas IPv6 du tout, sourds que sont leurs constructeurs chinois aux sirènes de la modernité.

Enfin, les problèmes ne sont pas forcément avec les matériels et logiciels du réseau local. Parfois, le problème est situé chez le pair avec qui on veut communiquer comme lorsque bit.ly (le RFC cite ce cas mais sans mentionner le nom comme si l'information n'était pas déjà publique...) avait publié un enregistrement AAAA invalide : NAT64 ne peut rien dans ce cas puisque la passerelle pense que le service est accessible en IPv6. C'est l'occasion de rappeler qu'il vaut mieux ne pas publier de AAAA que d'en publier un erroné.

On l'a vu, une bonne partie de la connectivité externe des deux réseaux de test dépendait de NAT64, puisqu'une grande partie de l'Internet n'est hélas pas joignable en IPv6. La section 6 tire le bilan spécifique de ce protocole et il est très positif : pas de problèmes de fond, juste quelques bogues dans l'implémentation, corrigées au fur et à mesure.

Le cas des adresses IPv4 littérales dans les URL est irrémédiable (le DNS n'est pas utilisé donc DNS64 ne peut rien faire) mais rare. Les deux seuls cas bloquants concernaient certaines pages YouTube (tiens, pourquoi est-ce que les erreurs de bit.ly sont mentionnées sans indiquer son nom, alors que YouTube est désigné dans le RFC ?) et une page de réservation d'un hôtel qui renvoyait vers un URL contenant une adresse IPv4. Les auteurs du RFC ont mesuré les 10 000 premiers sites Web du classement Alexa et trouvé que 0,2 % des 1 000 premiers ont un URL IPv4 dans leur page (pour charger du JavaScript, du CSS ou autre), et que ce chiffre monte à 2 % pour les 10 000 premiers (ce qui laisse entendre que les premiers sont mieux gérés). Cette stupide erreur n'empêchait pas le chargement de la page, elle privait juste le lecteur d'un bandeau de publicité ou d'une image clignotante. Ce n'est donc pas un problème sérieux en pratique.

Les auteurs ont également testé avec wget le chargement des pages d'accueil de ces 10 000 sites en comparant un accès IPv4, un accès IPv6 sans NAT64 et le réseau de test, IPv6 pur mais avec NAT64. En IPv4 pur, 1,9 % des sites ont au moins un problème (pas forcément la page d'accueil elle-même, cela peut être un des composants de cette page), ce qui donne une idée de l'état du Web. Le RFC note toutefois que certains problèmes peuvent être spécifiques à ce test, si le serveur refuse à wget du contenu qu'il accepterait de donner à un navigateur Web. wget avait été configuré pour ressembler à un navigateur (le RFC ne le dit pas mais je suppose que cela veut dire des trucs comme `--user-agent="Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:11.0) Gecko/20100101 Firefox/11.0"`). Mais cela ne suffit pas dans des cas comme l'absence d'une adresse pour un nom de domaine, qui ne gène pas le navigateur qui ajoute `www` devant à tout hasard, chose que ne fait pas wget.

Avec le réseau IPv6 pur, 96 % des sites échouent (ce qui est logique, vu le nombre de sites Web accessible en v6). Les sites Google étaient une des rares exceptions.

Avec le réseau IPv6 aidé de la passerelle NAT64, le taux d'échec est de 2,1 %, quasiment le même qu'en IPv4 pur (la différence venant surtout des adresses IPv4 littérales).

Sur quoi doivent porter les efforts de mesure futurs (section 7) ? Le RFC estime important de mesurer plus précisement les phénomènes à l'œuvre lors d'une connexion utilisant NAT64. Certains utilisateurs ont signalé des ralentissements, mais qui ne sont pas confirmés par une analyse des paquets et des temps de réponse. S'il se passe quelque chose qui ralentit, c'est plus subtil, et cela devrait être investigué.

Compte-tenu de cette expérience, quelles conclusions en tirer (section 8) ? Comme indiqué plus haut, la principale est qu'un réseau purement IPv6 est viable. (Le RFC ne le précise pas mais je rajoute : s'il est géré par un informaticien compétent et disponible. Cette expérience n'est pas encore reproductible par M. Toutlemonde.)

Seconde grande conclusion : il reste du travail, trop de petites bogues sont encore présentes, et à beaucoup d'endroits.

Bref, aujourd'hui, il reste prudent d'utiliser plutôt la double-pile (IPv6 **et** IP4) pour un réseau de production. Un réseau IPv6 pur, à part pour le "*geek*", est surtout intéressant pour des environnements très contrôlés, comme par exemple celui d'un opérateur de téléphonie mobile qui fournirait un modèle de téléphone obligatoire, permettant de s'assurer que tous les clients aient ce qu'il faut.

Comme tout ne s'arrangera pas tout seul, nos héroïques explorateurs revenant de la terre lointaine et mystérieuse où tout ne marche qu'en IPv6 suggèrent des actions à entreprendre pour se préparer à notre future vie dans ce monde :

— La découverte du résolveur DNS reste un des points qui marchent le plus mal, probablement parce qu'elle fonctionne bien en double-pile, masquant les problèmes IPv6. Par exemple, l'utilisation des annonces RA du RFC 6106 devrait être systématique et c'est loin d'être le cas, par exemple dans les divers systèmes utilisant Linux.
— Autre problème, les divers "*network managers*", ces logiciels qui règlent le ballet des différentes composants de l'accès au réseau : même lorsque le système dispose de tous les composants qui marchent en IPv6, le "*network manager*" ne les utilise pas toujours correctement. En essayant Ubuntu, par exemple, il est clair que ce système n'a jamais été testé dans un environnement purement IPv6.
— Les pare-feux ont souvent des capacités IPv6 inférieures à celles qu'ils ont en v4.
— Plusieurs applications que les utilisateurs considèrent comme très importantes (les jeux vidéo, par exemple) ne marchent pas du tout en IPv6 : du travail pour les programmeurs, qui devraient au miminum utiliser des API qui sont indépendantes de la version d'IP.

Le RFC note que pratiquement aucune de ces actions ne nécessite d'action dans le champ de la normalisation. Cela veut dire que toutes les normes nécessaires sont là, l'IETF a terminé son travail et c'est maintenant aux programmeurs et aux administrateurs réseaux d'agir.