RFC 6346 : The A+P Approach to the IPv4 Address Shortage

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 août 2011

Date de publication du RFC : Août 2011

http://www.bortzmeyer.org/6346.html

Pendant longtemps, le problème technique posé à l'IETF par l'épuisement des adresses IPv4 <http://www.bortzmeyer.org/epuisement-adresses-ipv4.html> était traité par la perspective d'une migration vers IPv6, et les efforts de l'IETF allaient vers la création de jolis mécanismes de transition <http://www.bortzmeyer.org/transition-ipv6-guilde.html> vers ce nouveau protocole. Mais la migration est très loin d'être terminée (sur beaucoup de sites, elle n'a même pas réellement commencé) et, dans des régions comme l'Asie, les adresses IPv4 ne sont plus seulement difficiles à obtenir et chères, elles sont tout simplement toutes utilisées. Il faut donc maintenant, en urgence, mettre au point des mécanismes qui permettront de vivre pas trop mal en attendant le déploiement d'IPv6. C'est le cas du mécanisme **A+P** ("*Address plus Port*") de ce RFC.

Tous ces mécanismes sont pour le **court terme**. À moyen et à long terme, la solution correcte, comme avant, reste le passage à IPv6. Toutefois, pour tous les paresseux et les irresponsables qui n'ont pas encore commencé cette migration, il est trop tard : même s'ils se réveillaient demain, ils n'auraient pas le temps de terminer leur transition vers IPv6 avant que le manque d'adresses IPv4 ne devienne un vrai blocage pour leur activité.

A+P, présenté dans ce RFC, propose une solution pour limiter les dégâts, chez les derniers utilisateurs d'IPv4. Cet article va être relativement court, par manque de temps pour explorer en détail cette technologie, mais aussi parce que je suis sceptique : c'est très gentil d'essayer de maintenir la tête des utilisateurs IPv4 hors de l'eau encore quelques mois ou quelques années, mais cela va coûter pas mal d'efforts, qui seraient mieux employés à leur apprendre à nager (à déployer IPv6).

Donc, quels sont les principes d'A+P ? Comme les miracles n'existent pas, l'idée de base est un compromis. On va sacrifier quelques bits du numéro de port pour les donner à l'adresse IP. Ce faisant, on réduit la capacité des applications à allouer beaucoup de connexions ouvertes (chacune nécessitant un port) mais il n'y a plus guère le choix. Comme le note le RFC, « "*the need for addresses is stronger than the*

need to be able to address thousands of applications on a single host" » ou, en termes plus brutaux, « en cas de famine, on ne réclame pas d'assaisonnement sur tous les plats ».

Cette famine fait que le partage d'adresses IPv4, déjà largement réalisé au sein d'une même maison ou entreprise, va forcément s'étendre et que des clients différents, sans lien entre eux, partageront la même adresse IP. L'idée d'A+P est que, même si l'adresse ne sera plus unique, le couple {adresse, partie du port} restera unique par client. Avec 65536 ports possibles, on peut mettre 65536 clients sur une même adresse IP (si chacun se contente d'un seul port), 256 (avec 256 ports chacun), ou un seul (avec le système actuel où le client a 65536 ports)... L'un des intérêts d'A+P est qu'il limite (sans toutefois le supprimer) le recours au NAT et à tous ses inconvénients.

En effet, une alternative à A+P serait de déployer des gros routeurs NAT (CGN, pour "*Carrier-grade NAT*") dans les réseaux des FAI, routeurs qui traduiraient pour plusieurs clients (au contraire du routeur NAT typique d'aujourd'hui, qui n'opère que pour un seul client). La section 1.1 explique pourquoi c'est une mauvaise idée : le CGN ne maintient pas la connectivité de bout en bout, qui est au cœur d'Internet. Par exemple, le déploiement d'une nouvelle application qui échange des adresses serait bloqué tant que le routeur CGN (qui est sous le contrôle du FAI) n'est pas mis à jour avec un nouvel ALG. Les techniques qui permettent de rendre les routeurs NAT actuels un peu moins insupportables (configuration manuelle de la redirection de port, UPnP) ne marchent en effet pas sur un CGN. Enfin, les routeurs CGN stockent un état considérable (des centaines ou des milliers de clients) et sont donc un point de faiblesse du réseau : si un routeur CGN redémarre, tout est perdu.

La section 3 détaille les contraintes que s'impose A+P, afin d'éviter ces défauts :
— Maintien, dans la mesure du possible, du principe de bout en bout (ne pas utiliser l'épuisement des adresses v4 comme excuse pour limiter l'usage que les utilisateurs peuvent faire de l'Internet),
— Compatibilité avec l'existant, par exemple, les anciennes applications doivent marcher comme avant,
— Ne pas garder trop d'état dans le réseau du FAI,
— Traçabilité des utilisateurs, c'est-à-dire capacité à fliquer quel ordinateur a fait telle action,
— Ne pas décourager la migration vers IPv6, comme, dit le RFC, le fait le NAT444 <`http://www.bortzmeyer.org/nats.html`> (ou « double-NAT »).

Ce cahier des charges est-il réaliste ? Le reste de la section 3 raconte comment fonctionne A+P. Il y a trois fonctions :
— Traduction : pour gérer les anciennes applications, qui allouent des ports sans restriction, et tournent sur une machine qui imagine être la seule au monde à avoir cette adresse IP, il faudra une fonction de traduction analogue au NAT actuel,
— Encapsulation et décapsulation : s'il y a des anciens routeurs sur le trajet, il faudra encapsuler les paquets. Ce sera le travail du PRR ("*Port Range Router*"),
— Signalisation : les machines nouvelles (qui connaissent A+P), ont besoin de savoir quelle plage de ports leur a été allouée.

Le traducteur devra faire bien attention, contrairement au NAT actuel, à n'utiliser en sortie que des ports appartenant à la plage allouée.

La signalisation est le gros morceau (section 3.3.1). A+P peut utiliser plusieurs protocoles pour cela (à ma connaissance, tous sont encore au stade de brouillon). Le protocole doit permettre d'informer une machine A+P de l'adresse IPv4 publique à utiliser, des plages de ports réservées pour elle, de la durée pendant laquelle ces allocations restent valables, etc. L'allocation des ports peut être statique (« tu as de 20 000 à 31 000 ») ou dynamique par des nouveaux protocoles comme UpNPv2.

Si on a un site client réduit, avec une seule machine, qui connaît A+P, et un équipement A+P chez le FAI, et que l'adresse IPv4 publique donnée au site client est {`192.0.2.1`, ports 20 000 à 31 000}, un paquet sortant du site client est émis tel quel (la machine sait n'utiliser comme port source que ceux alloués) et le paquet rentrant est traité ainsi dans l'équipement A+P :

— En regardant le port de destination, on trouve à quel client est destiné ce paquet (routage par le port, l'adresse IP ne suffisant plus, car elle est partagée entre plusieurs clients),
— On lui transmet.

Dans ce cas, pas besoin de traduction. Évidemment, dans la réalité, ce sera plus compliqué : il y aura plusieurs machines sur le site du client et, surtout, ces machines ne connaîtront pas toutes A+P et émettront donc des paquets avec des ports source quelconques. Cette fois, on ne pourra pas se passer de traduction : une machine chez le client (et non pas chez le FAI) va alors traduire ces paquets pour mettre l'adresse source allouée et changer le port source pour un des ports alloués. Au retour, ce même traducteur fera l'opération inverse, très proche de celle des routeurs NAT d'aujourd'hui.

Est-ce que tout cela marche vraiment en pratique ? La section 3.4 rend compte des essais effectués à France Télécom. Les plus gros problèmes sont survenus avec des mises en œuvre de BitTorrent, qui veulent accepter des connexions entrantes sur un port précis mais, globalement, il n'y a pas eu trop de problèmes.

Si vous aimez les considérations pratiques, la section 5 est vouée aux problèmes de déploiement. Par exemple, pour des sites client en ADSL ou connectés par le câble, le modèle le plus logique est que les fonctions de traduction soient dans la "*box*", le FAI allouant les plages de ports et les communiquant audit CPE. (Pour un réseau 3G, c'est plus complexe, car le "*smartphone*" ne peut pas forcément assurer beaucoup de fonctions.)

Comment le FAI décide-t-il de l'allocation des ports (section 5.2) ? Une solution évidente est de diviser équitablement. Si on met cent clients par adresse IP, chacun a 655 ports et se débrouille avec. Le risque de cette allocation statique est alors que les clients les plus gourmands soient limités, alors que les autres ont des ports libres. L'idéal serait donc que l'allocation soit dynamique, s'ajustant à la demande.

Comme avec le NAT, certains protocoles vont poser des problèmes particuliers. Ainsi, ICMP (section 5.3.2), n'ayant pas le concept de port, va être difficile à faire passer. Pour les messages d'erreur transportés par ICMP, comme ils incluent le début du paquet qui a provoqué l'erreur, on peut y trouver un numéro de port et router par ce biais. Pour les messages d'écho (ceux qu'utilise ping), il n'y a pas d'autre solution que de bricoler avec des champs comme l'identificateur du message ICMP, ou comme le numéro de séquence ICMP, en les utilisant comme numéros de port.

Parmi les autres problèmes concrets que devra résoudre la machine A+P, il y a enfin la fragmentation, source traditionnelle d'ennuis. Ici, le routage se faisant en fonction du port de destination, un réassemblage des fragments entrants est nécessaire, pour déterminer ledit port.

A+P, on l'a vu, est un système complexe. Il a quelques limitations, documentées en section 5.3.4. La principale est que les ports bien connus, notamment le célébrissime port 80, utilisé par les serveurs HTTP, ne seront pas forcément disponibles. Si le client qui a obtenu une plage de ports située au début (du côté des ports bien connus, en dessous de 1024) est heureux (et on peut même envisager de le faire payer plus cher pour ce « privilège »), les autres n'ont pas de solution, s'ils veulent faire tourner un serveur HTTP sur leur machine.

Enfin, A+P marche mal ou pas du tout avec IPsec, bien qu'il soit possible de contourner le problème avec les techniques du RFC 3715 [1].

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc3715.txt`

Et question sécurité, ça se passe comment ? La section 7 fait la liste des questions de sécurité liées à A+P. La première est la difficulté de suivre à la trace un utilisateur. Pour savoir qui a osé commettre un délit aussi grave que de partager de la culture <http://www.numerama.com/magazine/19648-tous-ces-delits-juges-moins-graves-que-le-partage-de-la-culture.html>, il faut que l'espion qui surveille le trafic ait noté le numéro de port source (cf. RFC 6302), et que l'équipement A+P ait enregistré qui avait la plage où se trouve ce numéro. A+P est donc plus efficace pour le flicage qu'un CGN ou qu'un NAT traditionnel, qui doit noter une information bien plus évanescente (qui avait ce port à ce moment-là). Reste à savoir si c'est un avantage...

Quel est l'état des implémentations de A+P au moment de la publication de ce RFC ? L'ISC a A+P dans son routeur logiciel (surtout connu pour DS-Lite) AFTR <http://www.isc.org/software/aftr> (qui ne semble actuellement pourvoir faire que de l'allocation de ports statique), France Télécom / Orange a également un logiciel libre en <http://opensourceaplusp.weebly.com/>, et il existe le logiciel 4RD de la société ipinfusion.com. Pour le futur, Iskratel et, dans une moindre mesure, Cisco et Juniper, ont indiqué qu'ils y travaillaient.