RFC 7269 : NAT64 Deployment Options and Experience

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 20 juin 2014

Date de publication du RFC : Juin 2014

https://www.bortzmeyer.org/7269.html

L'épuisement des adresses IPv4 <https://www.bortzmeyer.org/epuisement-adresses-ipv4.html> étant allé plus vite que le déploiement d'IPv6 (qui est freiné par la passivité de nombreux acteurs), il y a de plus en plus souvent le problème de connecter des réseaux modernes, entièrement en IPv6, à des machines anciennes restées en IPv4. La solution de l'IETF à ce problème est le déploiement de NAT64, normalisé dans le RFC 6144 [1]. Ce nouveau RFC documente l'expérience concrète de trois gros opérateurs avec NAT64. Qu'est-ce qui a marché, qu'est-ce qui a raté ?

Ces réseaux entièrement IPv6 ont l'avantage de ne pas avoir de problème de manque d'adresses IP (même les préfixes du RFC 1918 peuvent être insuffisants pour un grand réseau) et de permettre d'utiliser un plan d'adressage propre dès le début (pas de contorsions pour faire durer les adresses) et une architecture unique (un seul protocole à gérer). Pour les réseaux mobiles, par exemple, cela signifie un seul contexte PDP, ce qui simplifie les choses. Mais, comme vu plus haut, leur inconvénient est qu'ils ne peuvent pas accéder aux services qui restent archaïquement en IPv4 seul comme Twitter ou impots.gouv.fr. Et, même si les professionnels sérieux ont compris depuis longtemps l'importance de migrer vers IPv6, les résistances des plus attardés vont sans doute durer longtemps et on peut penser que des réseaux uniquement IPv4 seront encore en fonctionnement pendant de longues années. Au lieu du plan de transition envisagé au début (« tout le monde en IPv4 » -¿ « double-pile - IPv4 et IPv6 - déployée progressivement » -¿ « tout le monde en IPv6 »), il faut maintenant travailler dans le cadre d'un nouveau plan, « tout le monde en IPv4 » -¿ « certains réseaux en v4 seul et certains en v6 seul » -¿ « tout le monde en IPv6 ». L'étape intermédiaire a été décrite dans les RFC 6145 et RFC 6146 et son composant le plus connu est nommé **NAT64**. NAT64 effectue une traduction d'adresses de IPv6 vers IPv4 en sortie du réseau moderne vers le réseau archaïque et l'inverse lorsque la réponse arrive.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6144.txt

NAT64 a donc une partie des inconvénients du NAT IPv4 traditionnel (NAT44), comme le montre le compte-rendu d'expérience du RFC 6586 mais il parait incontournable, sauf si le déploiement d'IPv6 s'accélérait subitement dans l'Internet, ce qu'on ne voit pas venir.

30 % des opérateurs, selon un sondage signalé dans le RFC 6036 prévoient de déployer NAT64 ou un système équivalent. NAT64 est également systématiquement présent lors des réunions des organismes de normalisation ou de gestion des ressources réseau, comme l'IETF ou le RIPE. À chaque réunion, un des réseaux WiFi est « IPv6 seulement » et bénéficie d'une passerelle NAT64 vers l'extérieur. De nombreux experts peuvent ainsi tester en vrai les technologies modernes. Bref, NAT64 est désormais une technique importante et il est donc nécessaire d'étudier ses conséquences pratiques. C'est le rôle de ce nouveau RFC, qui ne décrit pas un nouveau protocole mais offre simplement un retour d'expérience.

À noter que NAT64 peut être déployé en deux endroits :
— Côté FAI, ce qu'on appelle le NAT64-CGN (pour "*Carrier-Grade NAT*"), sous forme d'un gros traducteur NAT dans le réseau du FAI,
— Ou bien côté fournisseur de services/contenus, ce qu'on appelle le NAT64-FE (pour "*Front End*"), sous forme d'un traducteur NAT dans le "*data center*", devant les serveurs.

Le premier cas est celui du FAI tout en IPv6 mais qui veut permettre à ses clients d'accéder à des services IPv4. Le second est celui du fournisseur de services ou de contenus dont l'infrastructure est malheureusement entièrement ou partiellement en IPv4 mais qui veut fournir un accès IPv6, en ne gérant qu'un traducteur, au lieu de mettre à jour tous ses systèmes. (Le RFC ne discute pas la question de savoir si c'est une idée intelligente ou pas...)

À noter que le cas d'un NAT64 chez l'utilisateur (le célèbre M. Michu), dans son CPE, n'est pas mentionné. Un NAT64 dans le CPE imposerait un réseau du FAI qui soit en IPv4 jusqu'au client, justement ce qu'on voudrait éviter. Le RFC recommande plutôt de mettre les traducteurs NAT64 près de la sortie du FAI, de façon à permettre à celui-ci de conserver un réseau propre, entièrement IPv6 (section 3.1.3). Certes, cela va concentrer une bonne partie du trafic dans ces traducteurs mais NAT64 est fondé sur l'idée que la proportion de trafic IPv4 va diminuer avec le temps : les traducteurs n'auront pas besoin de grossir. Aujourd'hui, 43 des cent plus gros sites Web (selon Alexa) ont déjà un AAAA et court-circuitent donc complètement le NAT64.

Autre problème avec la centralisation, le risque de SPOF. Si le CPE de M. Michu plante, cela n'affecte que M. Michu. Si un gros traducteur NAT64, gérant des milliers de clients, plante, cela va faire mal. Il faut donc soigner la disponibilité (voir section 4).

Centraliser dans un petit nombre de traducteurs près de la sortie évite également d'avoir plusieurs équipements NAT64, avec les problèmes de traçabilité que cela poserait, par exemple s'ils ont des formats de journaux différents.

NAT64 nécessite un résolveur DNS spécial, utilisant DNS64 (RFC 6147). En effet, les clients, n'ayant qu'IPv6, ne demanderont que des enregistrements AAAA (adresses IPv6) dans le DNS. Le résolveur DNS64 doit donc en synthétiser si le site original (par exemple `twitter.com`) n'en a pas. Dans certains cas, par exemple une page Web où les liens contiennent des adresses IPv4 littérales (comme `<a href="http://192.0.2.80/">`), la solution 464xlat du RFC 6877 permet de se passer de DNS64. Mais, dans la plupart des déploiements NAT64, DNS64 sera indispensable.

NAT64 va probablement coexister avec NAT44 (le NAT majoritaire aujourd'hui `<https://www.bortzmeyer.org/nats.html>`), les réseaux locaux chez l'utilisateur gardant un certain nombre de machines IPv4. Si le RFC 6724 donne par défaut la priorité à IPv6, quelque soit le type de connectivité IPv6, les algorithmes comme celui des globes oculaires heureux (RFC 6555) peuvent mener à préférer

IPv6 ou IPv4 et même à en changer d'un moment à l'autre, en fonction de leurs performances respectives. Ces changements ne seront pas forcément bons pour l'expérience utilisateur.

Pour le NAT64-FE (juste en face des serveurs), le RFC 6883 donnait quelques indications, qui peuvent s'appliquer à NAT64. Ici, DNS64 n'est pas forcément nécessaire, les adresses IPv6 servant à adresser les serveurs sont en nombre limité et peuvent être mises dans le DNS normal.

La section 4 de notre RFC revient sur l'exigence de haute disponibilité, essentielle si on veut que le passage de IPv4+IPv6 à IPv6+NAT64 ne diminue pas la fiabilité du service. Je me souviens d'une réunion RIPE où il n'y avait qu'un seul résolveur DNS64 et, un matin, il avait décidé de faire grève. DNS64 étant indispensable à NAT64, les clients étaient fort marris. Mais la redondance des résolveurs DNS est un problème connu, et relativement simple puisqu'ils n'ont pas d'état. Le RFC se focalise surtout sur la haute disponibilité des traducteurs NAT64 car eux peuvent avoir un état (comme les traducteurs NAT44 d'aujourd'hui). Il y a trois types de haute disponibilité : le remplaçant froid, qui est un traducteur NAT64 inactif en temps normal, et qui ne se synchronise pas avec le traducteur actif. Il est typiquement activé manuellement quand le traducteur actif tombe en panne. Comme il n'a pas de copie de l'état (la table des sessions en cours, aussi nommée BIB pour "*Binding Information Base*"), toutes les connexions casseront et devront être réétablies. Le second type est le remplaçant tiède. Il ne synchronise pas les sessions mais il remplace automatiquement le traducteur NAT64 principal, par exemple grâce à VRRP (RFC 5798). Au cours des tests faits par les auteurs du RFC, il fallait une minute pour que VRRP bascule et que les trente millions de sessions de la BIB soient établies à nouveau. Enfin, il y a le remplaçant chaud, qui synchronise en permanence l'état des sessions. Cette fois, la bascule vers le remplaçant est indétectable par les clients. Pour s'apercevoir rapidement de la panne du traducteur principal, le test a utilisé BFD (RFC 5880) en combinaison avec VRRP. Cette fois, la bascule des trente millions de sessions n'a pris que trente-cinq millisecondes, soit une quasi-continuité de service. Évidemment, cela nécessite du logiciel plus complexe dans les traducteurs, surtout vu l'ampleur de la tâche : avec 200 000 utilisateurs, il faut créer et détruire environ 800 000 sessions par seconde ! Un lien à 10 Gb/s entre les traducteurs n'est pas de trop, pour transporter les données nécessaires pendant les pics de trafic.

Alors, remplaçant froid, tiède ou chaud ? Notre RFC ne tranche pas définitivement, disant que cela dépend des applications. (L'annexe A du RFC contient des tests spécifiques à certaines applications.) Environ 90 % du trafic pendant les tests était dû à HTTP. Ce protocole n'a pas besoin de continuité de service, chaque requête HTTP étant indépendante. Le traducteur chaud n'est donc pas indispensable si HTTP continue à dominer (au passage, c'est pour une raison analogue que la mobilité IP n'a jamais été massivement utilisée). En revanche, le "*streaming*" souffrirait davantage en cas de coupure des sessions. (Curieusement, le RFC cite aussi le pair-à-pair comme intolérant aux coupures de session, alors que les clients BitTorrent, par exemple, s'en tirent très bien et réétablissent automatiquement des sessions.)

Le trafic d'un traducteur NAT64-CGN peut être intense et, même en l'absence de pannes, il est raisonnable de prévoir plusieurs traducteurs, afin de répartir la charge. Attention, si on fait du NAT64 avec état (le plus courant), il faut s'assurer que les paquets d'une session donnée arrivent toujours au même traducteur. Une façon de distribuer les requêtes entre les traducteurs, pour le NAT64-CGN, est de demander au serveur DNS64 de synthétiser des préfixes différents selon le client, et d'envoyer les différents préfixes à différents traducteurs, en utilisant le classique routage IP. Pour le NAT64-FE, les techniques classiques des répartiteurs de charge (condensation de certains paramètres puis répartition en fonction du condensat) conviennent.

La section 5 du RFC se penche ensuite sur le problème de l'adresse IP source. À partir du moment où on fait du NAT (quel que soit le NAT utilisé), l'adresse IP source vue par le destinataire n'est pas celle de l'émetteur mais celle du traducteur NAT. Cela entraine tout un tas de problèmes. D'abord, la traçabilité. Un serveur voit un comportement illégal ou asocial (par exemple, pour un serveur HTTP, des requêtes identiques coûteuses répétées, pour réaliser une attaque par déni de service). Son administrateur note

l'adresse IP du coupable. Qu'en fait-il ? Pour pouvoir remonter de cette adresse IP au client individuel, il faut garder les journaux du traducteur NAT (et aussi, le RFC l'oublie, que le serveur note l'heure exacte et le port, comme rappelé par le RFC 6302). Les auteurs du RFC ont tenté l'expérience avec 200 000 clients humains pendant 60 jours. Les informations sur les sessions étaient transmises par syslog (RFC 5424) à une station d'archivage. Avec toutes les informations conservées (heure, protocole de transport, adresse IPv6 source originelle - et son port, adresse IPv4 source traduite - et son port), soit 125 octets, les 72 000 sessions par seconde ont produit pas moins de 29 téra-octets de données. La seule activité de journalisation, pour faire plaisir au CSA, nécessite donc une infrastructure dédiée (machines, disques et liens) et coûteuse.

Le système pourrait toutefois être amélioré. Au lieu de noter le port pour chaque session, on pourrait allouer une plage de ports à chaque client et réduire ainsi le volume à journaliser (seule l'allocation de la plage serait notée ; ceci dit, cela compliquera sérieusement le logiciel de recherche, qui aura moins de données à fouiller mais davantage de calculs à faire). On peut même statiquement allouer des plages de port à chaque client, supprimant ainsi une grande partie de la journalisation. Mais attention, ces méthodes d'optimisation ont deux gros défauts : elles diminuent l'efficacité du multiplexage (certains client vont tomber à court de ports alors que d'autres en auront des inutilisés) et elle est contraire aux recommandations de sécurité du RFC 6056, qui prescrit des ports sources difficiles à deviner, pour compliquer la tâche d'un attaquant.

Autre problème lié à la traduction d'adresses, la géolocalisation. Un serveur qui géolocalise en se fondant sur l'adresse IP source trouvera la position du traducteur, pas celle de l'utilisateur. Parfois, cela peut être gênant. Le RFC 6967 cite quelques solutions possibles, mais aucune n'est parfaite : on peut par exemple se servir d'informations applicatives comme l'en-tête `Forwarded:` de HTTP (RFC 7239), ce qui ne marche évidemment qu'avec ce protocole.

La section 6 revient vers les problèmes de M. Michu, heureux utilisateur, sans le savoir, d'un traducteur NAT64. Quelle va être la qualité de son vécu d'utilisateur ? Ou, dit plus trivialement, est-ce que ça va marcher ou planter ? Certaines applications fonctionnent facilement et simplement à travers n'importe quel NAT <`https://www.bortzmeyer.org/nats.html`>, comme HTTP ou SSH. D'autres nécessitent un relais, une ALG. Les traducteurs NAT64 fournissent en général une ALG pour FTP (RFC 6384). Même pour les protocoles qui marchent « tout seuls » à travers NAT64, une ALG peut être utile pour la traçabilité, par exemple insérer des en-têtes `Via:` dans une requête HTTP ou `Received:` dans un message transmis en SMTP.

En testant des applications réelles, comme l'avait fait le RFC 6586, les auteurs de ce RFC ont trouvé que plusieurs continuent à planter, ce qui plaiderait pour l'ajout à NAT64 des fonctions 464xlat du RFC 6877. Ainsi, SIP n'a pas réussi à appeler depuis la machine IPv6 pure vers un "*softphone*" en IPv4 pur (les adresses IP sont transmises dans la signalisation, et ne sont pas traduites par NAT64). Il faut donc, comme recommandé par le RFC 6157, que les clients SIP utilisent ICE (RFC 8445), qui a l'avantage de traiter également le problème des pare-feux.

Pour IPsec, le bilan est également mitigé : comme prévu, AH n'a pas fonctionné (puisque son but est d'empêcher toute modification des en-têtes, donc des adresses IP source et destination), mais ESP non plus, les paquets du protocole de couche 4 50, le protocole d'ESP, n'étant pas traduits. Utiliser ESP sur UDP, comme décrit dans le RFC 3947, aurait probablement résolu ce problème.

Le tableau de la section 6.1 résume les applications testées et les résultats :
— Web : pas de problème, sauf les rares cas où un URL inclut une adresse IPv4 littérale,
— Messagerie instantanée : la plupart des services testés ont échoué (mon expérience personnelle est que XMPP et IRC fonctionnent très bien à travers NAT64, donc je suppose qu'il s'agissait de protocoles fermés et de logiciels privateurs),

— Jeux : les jeux non-Web ont presque tous échoué,
— SIP : échec,
— IPsec : échec (vor ci-dessus),
— Partage de fichiers en pair-à-pair : échec, par exemple avec eMule,
— Courrier électronique : pas de problème,
— FTP : pas de problème.

L'annexe A décrit en outre les délais acceptables pour ces applications, lorsque le traducteur NAT64 tombe en panne et est remplacé. Les jeux interactifs, par exemple, ne supportent pas les longues interruptions.

Comme un traducteur NAT64 gène certains services, il serait utile de fournir aux clients un moyen de contrôler, par exemple, l'ouverture de ports entrants, les ports sortants affectés, etc. Il existe un protocole standard pour cela, PCP (RFC 6887), qui serait un ajout très utile aux traducteurs NAT64 mais, pour l'instant, aucun n'en dispose encore.

La section 7 du RFC se penche ensuite sur les problèmes de MTU. Certes, il n'y a pas de tunnel avec NAT64, mais il peut y avoir des liens IPv4 dont la MTU est inférieure à 1 280 octets, le minimum des liens IPv6. Il n'y a pas de moyen simple, en NAT64-CGN, de gérer ce problème.

Bon, et la sécurité ? La section 9 étudie la question. NAT64 fait du suivi des sessions TCP et est donc normalement à l'abri de pas mal de types d'attaques par déni de service. On peut renforcer sa sécurité en déployant RPF (RFC 3704), pour empêcher les usurpations d'adresses IPv6.

Pour NAT64-FE, une attaque possible serait de remplir la table des sessions en commençant plein de sessions vers les serveurs situés derrière le traducteur NAT64-FE. Celui-ci doit donc avoir un mécanisme de limitation de trafic.

Ah, et puis il faut rappeler qu'un résolveur DNS64, étant un résolveur menteur (pour la bonne cause, certes, mais menteur tout de même), s'entend mal avec DNSSEC. Le RFC 6147 contient de nombreuses notes à ce sujet.

Ce RFC est issu d'un processus qui a commencé par un exposé à l'IETF <http://www.ietf.org/proceedings/82/slides/v6ops-5.pdf>. Il y a eu quelques objections de principe (un certain nombre de participants à l'IETF estiment que toute traduction est une erreur, NAT64 comme les autres).