RFC 8781 : Discovering PREF64 in Router Advertisements

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 25 avril 2020

Date de publication du RFC : Avril 2020

https://www.bortzmeyer.org/8781.html

Lors qu'on utilise le système NAT64, pour permettre à des machines situées sur un réseau purement IPv6 de parler avec des machines restées en seulement IPv4, il faut utiliser un résolveur DNS spécial ou bien connaitre le préfixe utilisé pour la synthèse des adresses IPv6. Ce nouveau RFC décrit une option RA ("*Router Advertisement*") pour informer les machines du préfixe en question.

NAT64 est normalisé dans le RFC 6146[1] et son copain DNS64 dans le RFC 6147. Il permet à des machines situées dans un réseau n'ayant qu'IPv6 de parler à des machines restées uniquement sur le protocole du siècle dernier, IPv4. Le principe est que, pour parler à la machine d'adresse `192.0.2.1`, la machine purement IPv6 va écrire à `64:ff9b::192.0.2.1` (soit `64:ff9b::c000:201`.) C'est le routeur NAT64 qui changera ensuite le paquet IPv6 allant vers `64:ff9b::192.0.2.1` en un paquet IPv4 allant vers `192.0.2.1`. NAT64 est, par exemple, largement utilisé dans les grandes réunions internationales comme l'IETF ou le FOSDEM, où le réseau WiFi par défaut est souvent purement IPv6. Mais pour que la machine émettrice pense à écrire à `64:ff9b::192.0.2.1`, il faut qu'elle connaisse le préfixe à mettre devant les adresses IPv4 (`64:ff9b::/96` n'est que le préfixe par défaut.) La solution la plus courante, DNS64, est que le résolveur DNS mente, en fabriquant des adresses IPv6 pour les machines n'en ayant pas, dispensant ainsi les machines du réseau local IPv6 de tout effort. Mais si l'émetteur veut faire sa résolution DNS lui-même, ou bien utiliser un autre résolveur, qui ne fabrique pas les adresses v6 ? Une alternative est donc que cet émetteur fasse lui-même la synthèse. Pour cela, il faut lui fournir le préfixe IPv6 du routeur NAT64.

Cela résoudra le problème de la machine qui ne veut pas utiliser le résolveur DNS64 :
— Car elle veut faire la validation DNSSEC toute seule comme une grande,
— ou car elle veut se servir d'un résolveur extérieur, accessible via DoT (RFC 7858) ou DoH (RFC 8484),

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc6146.txt`

— ou bien que l'administrateur du réseau local ne fournit tout simplement pas de résolveur DNS64,
— ou encore car elle veut pouvoir utiliser des adresses IPv4 littérales, par exemple parce qu'on lui a passé l'URL `http://192.0.2.13/`, dans lequel il n'y a pas de nom à résoudre,
— ou enfin parce qu'elle utilise 464XLAT (RFC 6877) pour lequel la connaissance du préfixe IPv6 est nécessaire.

Pourquoi envoyer cette information avec les RA ("*Router Advertisements*") du RFC 4861 plutôt que par un autre mécanisme ? La section 3 du RFC explique que c'est pour être sûr que le sort est partagé ; si le routeur qui émet les RA tombe en panne, on ne recevra plus l'information, ce qui est cohérent, alors qu'avec un protocole séparé (par exemple le PCP du RFC 7225), on risquait de continuer à annoncer un préfixe NAT64 désormais inutilisable. En outre, cela diminue le nombre de paquets à envoyer (l'information sur le préfixe NAT64 peut être une option d'un paquet RA qu'on aurait envoyé de toute façon.) Et, en cas de changement du préfixe, le protocole du RFC 4861 permet de mettre à jour toutes les machines facilement, en envoyant un RA non sollicité.

Enfin, l'utilisation des RA simplifie le déploiement, puisque toute machine IPv6 sait déjà traiter les RA.

L'option elle-même est spécifiée dans la section 4 du RFC. Outre les classiques champs Type (valeur 38 <`https://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xml#icmpv6-parameters-5`>) et Longueur (16 octets), elle comporte une indication de la durée de validité du préfixe, un code qui indique la longueur du préfixe (le mécanisme classique préfixe/longueur n'est pas utilisé, car toutes les longueurs ne sont pas acceptables), et le préfixe lui-même.

La section 5 explique comment les machines doivent utiliser cette option. Elle s'applique à tous les préfixes IPv4 connus. Si on veut router certains préfixes IPv4 vers un routeur NAT64 et d'autres préfixes vers un autre routeur, il faut quand même que les adresses IPv6 soient dans le même préfixe, et router ensuite sur les sous-préfixes de ce préfixe. Un autre problème découlant de ce choix d'avoir un préfixe IPv6 unique pour tout est celui des préfixes IPv4 qu'on ne veut pas traduire, par exemple des préfixes IPv4 purement internes. Il n'y a pas à l'heure actuelle de solution. De toute façon, l'option décrite dans notre RFC est surtout conçu pour des réseaux purement IPv6, et qui n'auront donc pas ce problème.

Attention, le préfixe IPv6 reçu est spécifique à une interface réseau. La machine réceptrice devrait donc le limiter à cette interface. Si elle a plusieurs interfaces (pensez par exemple à un ordiphone avec WiFi et 4G), elle peut recevoir plusieurs préfixes pour le NAT64, chacun ne devant être utilisé que sur l'interface où il a été reçu (cf. RFC 7556.)

Un petit mot sur la sécurité pour terminer. Comme toutes les annonces RA, celle du préfixe NAT64 peut être mensongère (cf. le RFC 6104 au sujet de ces « RAcailles ».) La solution est la même qu'avec les autres options RA, utiliser des techniques « "*RA guard*" » (RFC 6105.)

À noter qu'une annonce mensongère d'un préfixe IPv6 prévue pour le NAT64 affectera évidemment les adresses IPv4 qu'on tente de joindre (parfois au point de les rendre injoignables) mais pas les adresses IPv6, qui ne sont pas traduites et donc non touchées par cet éventuel RA tricheur. Comme une machine qui arrive à émettre et à faire accepter des RAcailles peut déjà facilement réaliser un déni de service, on voit que l'option de ce nouveau RFC n'aggrave pas les choses, en pratique.

Le RFC conclut que la procédure de validation des préfixes de la section 3.1 du RFC 7050 n'est pas nécessaire, si on a ce "*RA Guard*".

À l'heure actuelle, il ne semble pas que cette option soit souvent mise en œuvre, que ce soit dans les émetteurs ou dans les récepteurs. Mais Wireshark sait déjà le décoder <`https://twitter.com/alagoutte/status/1255404872117235712`>.