RFC 7217 : A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 2 mai 2014

Date de publication du RFC : Avril 2014

https://www.bortzmeyer.org/7217.html

Il existe actuellement de nombreuses méthodes (y compris manuelles) de générer les adresses IPv6 mais le mode SLAAC ("*Stateless Address Autoconfiguration*", normalisé dans le RFC 4862[1]) est particulièrement intéressant car il permet aux machines IPv6 de s'auto-configurer, sans qu'il soit nécessaire de mémoriser un état dans un serveur central. SLAAC fonctionne en concaténant à un **préfixe**, appris en écoutant les annonces des routeurs, un **identifiant d'interface** spécifique à la machine. Au tout début d'IPv6, il n'existait qu'une seule façon de générer ces identifiants d'interface, via l'adresse MAC de la machine. Puis une deuxième façon a été ajoutée, en tirant au sort l'identifiant d'interface, et en en changeant souvent, pour préserver sa vie privée (RFC 8981). Et une troisième, la plus sûre, où l'identifiant d'interface est dérivé d'une clé cryptographique (RFC 3972, mais ces CGA sont peu utilisés et notre RFC n'en parle guère). Notre nouveau RFC propose une quatrième façon, en condensant un secret, un certain nombre de caractéristiques de la machine et le préfixe, de manière à avoir des identifiants **stables** (comme ceux de la première façon), mais préservant quand même partiellement la vie privée (comme ceux de la deuxième façon) : l'identifiant d'interface change quand la machine change de réseau, ne permettant plus de la suivre à la trace. Cette méthode permet de choisir l'identifiant d'interface, elle ne modifie pas l'algorithme présenté dans le RFC 4862.

Les identifiants d'interface obtenus par l'adresse MAC simplifient bien des choses, notamment grâce à leur stabilité : on éteint la machine et on la rallume, on est sûr de retrouver la même adresse IPv6. Ainsi, lorsqu'on regarde le journal des connexions, on peut facilement retrouver la machine qu'on a repéré. Et créer des ACL est simple, puisque les adresses ne changent pas. Mais, en échange, ces identifiants ont des inconvénients, notamment pour la vie privée (cf. l'étude de l'IAB <http://www.iab.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc4862.txt

org/wp-content/IAB-uploads/2011/07/IPv6-addresses-privacy-review.txt>). La stabilité permet de suivre à la trace une machine pendant une longue période. Pire, même si la machine change de réseau, l'identifiant d'interface ne change pas, permettant de suivre cette trace même en cas de mobilité. D'autre part, comme l'adresse MAC dépend du constructeur, les adresses IPv6 d'un même parc ont donc souvent pas mal de bits en commun. Cette prévisibilité facilite notamment le balayage des adresses réseau (qui est normalement difficile en IPv6, vu le nombre d'adresses, cf. RFC 7707) et cette utilisation a été mise en œuvre dans des outils de reconnaissance existants (cf. mon article sur les attaques contre IPv6 <https://www.bortzmeyer.org/hacking-ipv6.html>). Enfin, des opérations de maintenance matérielle, comme de changer une carte Ethernet, vont faire changer l'adresse, annulant l'avantage de stabilité.

Le RFC 8981 voulait s'attaquer à ces problèmes avec ses adresses temporaires, où l'identifiant d'interface est choisi aléatoirement et change de temps en temps. L'idée est que la machine aurait une (ou plusieurs) adresses temporaires, une (ou plusieurs) adresses stables et qu'on utiliserait l'adresse temporaire pour les connexions sortantes, et l'autre pour les entrantes. Cela fournit une bonne protection, question vie privée, mais au prix de quelques inconvénients. Comme rien n'est gratuit en ce bas monde, ces adresses compliquent la vie de l'administrateur réseaux : interpréter le trafic qu'on voit passer est moins simple (beaucoup de techniques de protection de la vie privée ont ce défaut). On voit donc des entreprises couper ces adresses temporaires, afin de mieux surveiller leurs employés (cf. le récit de Ron Broersma <http://www.ipv6.org.au/10ipv6summit/talks/Ron_Broersma.pdf>). Les adresses privées peuvent également poser des problèmes pour les commutateurs réseaux <http://blog.bimajority.org/2014/09/05/the-network-nightmare-that-ate-my-week/>. En outre, pour des engins aux capacités limités (Internet des objets), la mise en œuvre d'une technique supplémentaire de génération des adresses peut être trop coûteuse. Enfin, comme les adresses traditionnelles sont toujours en place, certaines activités, comme le balayage d'adresses en suivant certains motifs, liés aux préfixes des constructeurs, restent possibles. Bref, la situation actuelle n'est pas complètement satisfaisante.

Notre nouveau RFC 7217 ne vise pas à remplacer les adresses temporaires, qui restent la bonne solution pour limiter le suivi d'une machine à la trace. Il veut plutôt remplacer les adresses basées sur l'adresse MAC, en gardant l'importante propriété de stabilité. L'idée est que les machines IPv6, dans le futur, auront uniquement une adresse stable, générée selon ce RFC, ou bien une adresse stable générée selon ce RFC et une adresse temporaire. L'adresse fondée sur l'adresse MAC aurait vocation à disparaître à terme (même si le RFC n'est pas aussi radical, cf. section 4, dernier élément de l'énumération), car elle facilite trop le balayage des réseaux et le suivi d'une machine mobile, sans avoir d'avantages par rapport à la solution de ce RFC.

L'objectif principal est le mécanisme SLAAC d'auto-configuration sans état mais la solution de ce RFC peut aussi s'appliquer à d'autres méthodes comme DHCP (RFC 8415), où le serveur peut utiliser un algorithme identique pour générer les adresses qu'il va distribuer.

La section 4 formalise le cahier des charges de la nouvelle méthode de génération des identifiants d'interface :

— Identifiant stable dans le temps, tant qu'on est connecté au même réseau (ce qui permet la corrélation entre les différentes activités de la machine dans le temps : si on ne veut pas cela, il faut utiliser les adresses temporaires),
— Identifiant dépendant du préfixe, de manière non prévisible par un observateur : si un client IPv6 contacte un serveur, puis change de réseau, puis contacte le même serveur, le serveur ne doit pas pouvoir facilement rapprocher les deux adresses (l'un des plus gros inconvénients de la méthode historique de génération d'adresses était cette possibilité de rapprochement),
— Identifiant non prévisible pour un observateur, même s'il connait des identifiants sur le même réseau,

— Il serait souhaitable, si possible, que l'identifiant soit indépendant de l'adresse MAC, de manière à ce qu'un changement de carte Ethernet ne se traduise pas par un changement d'adresss IP (les identifiants de la méthode historique sont trop stables pour préserver la vie privée mais changent trop lorsqu'on modifie le matériel).

À noter que l'algorithme exact utilisé pour le choix de l'identifiant d'interface (et donc de l'adresse) est une décision purement locale. Chaque machine fait comme elle veut, cela n'affecte pas l'interopérabilité. Sur un même réseau peuvent donc coexister des machines utilisant des algorithmes différents.

Je crois que j'ai un peu abusé de votre patience avec ces longs préliminaires. Peut-être auriez-vous préféré qu'on commence directement avec l'algorithme, la solution avant la description du problème ? En tout cas, arrivé à la section 5, vous avez cet algorithme. Il est trivial : l'identifiant d'interface est le résultat de l'application d'une fonction de condensation à plusieurs paramètres. Cette fonction doit donner un résultat qui, vu de l'extérieur, sera imprévisible, et elle ne doit pas être inversible. Un exemple d'une telle fonction est la fonction de condensation cryptographique SHA-256, dont on conservera les 64 bits de plus faible poids (par contre, MD5 ne doit pas être utilisé, cf. RFC 6151). Rappelez-vous que la génération de l'identifiant d'interface est un processus purement local et que différentes machines peuvent donc utiliser des fonctions de condensation différentes. Les paramètres de la fonction de condensation comportent au moins :

— Le préfixe des adresses IP (donc, changer de réseau fait changer d'identifiant d'interface, ce qui est souhaité), tel qu'on l'a appris, par exemple via les RA ("*Router Advertisement*"),
— Le nom de l'interface réseau, par exemple `eth0` pour une machine Linux, le but étant que la machine ait des adresses différentes pour chaque réseau, même s'ils utilisent le même préfixe,
— Un compteur des conflits (duplication d'adresses), qui part de zéro et est incrémenté chaque fois qu'une collision d'adresses (deux adresses identiques sur le même lien) a lieu,
— Une clé secrète, spécifique à la machine (tous les autres paramètres et, bien sûr, l'algorithme exact utilisé, peuvent être découverts plus ou moins facilement par un observateur : cette clé est donc indispensable si on veut des adresses vraiment imprévisibles),
— Et, facultativement, un identifiant du réseau, par exemple le SSID ou bien une des idées du RFC 6059.

L'annexe A du RFC fournit plusieurs idées pour le nom de l'interface réseau.On peut se servir de l'index d'interface (RFC 3493), du nom s'il est assez stable (un système d'exploitation où les interfaces réseau portent des noms dépendant du pilote utilisé aura des noms moins stables que Linux, où ce nom est générique, `eth1`, `wlan0`, etc), ou de tout autre identificateur fourni par le système (certains ont un UUID par interface réseau).

La clé secrète sera typiquement générée à l'initialisation de la machine, par un processus aléatoire ou pseudo-aléatoire, et stockée de manière à survivre aux redémarrages. Mais le RFC prévoit aussi le cas où elle pourrait être affichée, voire modifiée, par l'administrateur système. Le point important est de se rappeler que toute la sécurité de ce système en dépend donc elle doit être bien choisie.

Cet algorithme est largement inspiré de celui décrit dans le RFC 1948, pour un usage très différent.

Une fois obtenu un identifiant d'interface, il faut vérifier qu'il n'est pas dans les identifiants réservés du RFC 5453. Sinon, tous les bits de l'identifiant d'interface sont opaques (au sens du RFC 7136) et il ne faut donc pas leur chercher de signification particulière. Dernière vérification à faire avant d'utiliser l'adresse construite avec cet identifiant d'interface : tester qu'il n'y a pas de conflits avec une autre machine. En effet, comme chaque machine du réseau tire son identifiant d'interface au hasard, il y a une probabilité non nulle (mais très faible) que deux machines tirent le même identifiant et donc la même adresse IP. Elles doivent donc suivre la procédure de détection de conflits « DAD » du RFC 4862 (section 5.4).

C'est désormais la méthode de ce RFC qui est recommandée lorsqu'on veut des adresses stables (cf. RFC 8064), à la place des anciennes adresses dérivées de l'adresse MAC.

Il existait une mise en œuvre <`http://patchwork.ozlabs.org/patch/300938/`> pour Linux, par Hannes Frederic Sowa, mais, apparemment, cela a été intégré dans la distribution officielle peu après la sortie du RFC. (Par contre, à ma connaissance, rien pour Windows ou pour les BSD.)