RFC 7010 : IPv6 Site Renumbering Gap Analysis

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 14 septembre 2013

Date de publication du RFC : Septembre 2013

https://www.bortzmeyer.org/7010.html

Si on pouvait facilement renuméroter (changer les adresses IP) d'un réseau, beaucoup de problèmes seraient moins graves. Par exemple, il y aurait moins de demande pour des adresses PI. Mais, comme l'explique bien le RFC 5887 [1], renuméroter, en pratique, est **difficile**, et les administrateurs réseaux, à juste titre, font tout ce qu'ils peuvent pour l'éviter. Pour essayer de résoudre ce problème, l'IETF a un groupe de travail 6renum <http://tools.ietf.org/wg/6renum>, dont voici le troisième RFC. 6renum travaille à faciliter la rénumérotation en IPv6. Ce RFC 7010 est consacré à l'analyse des trous : que manque t-il comme protocole IETF pour faciliter la renumérotation ?

C'est un enjeu important pour l'Internet : sans renumérotation facile, les administrateurs réseaux se tourneront encore plus massivement vers les adresses PI, qui exercent une pression sur la table de routage (RFC 4984). Le RFC 6879, plus pratique, décrivait un certain nombre de scénarios de renumérotation et ce nouveau document, le RFC 7010 va bâtir sur ces scénarios en regardant ce qui gêne leur réalisation (et qui nécessitera un travail à l'IETF). Les machines ayant une adresse IP statique, typiquement les serveurs, sont traitées dans un autre document, le RFC 6866. De même, le cas d'une rénumérotation urgente et non planifiée, par exemple suite à la faillite d'un FAI, n'est pas couvert ici.

Donc, d'abord (section 2), qu'appelle t-on une rénumérotation réussie ? Que voudrait-on comme propriétés ? Une rénumérotation part d'un ancien préfixe et arrive à un nouveau. Pour les moyennes et grandes organisations, on a pu recevoir ce nouveau préfixe par des moyens humains (message électronique, par exemple). Pour les petites, on a pu le recevoir par une délégation, genre DHCPv6-PD ("*Prefix Delegation*", RFC 8415). Dans les deux cas, on voudrait plein de choses :

— Le nouveau préfixe devrait être reçu automatiquement et correctement,
— Les adresses IP dans ce nouveau préfixe devraient être acquises automatiquement par les machines, sans intervention humaine,

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5887.txt

— Les endroits où se trouvent configurées ces adresses devraient se mettre à jour,
— Même si ça se passe tout seul, on souhaiterait être notifié de ce changement et pouvoir le superviser,
— Enfin, on voudrait que la rénumérotation ne casse pas les sessions applicatives en cours (pensez à SSH...) Comme ce RFC ne regarde que les renumérotations contrôlées (pas celles faites en urgence, à l'arrache), cet objectif peut être atteint en ayant une période de transition qui soit plus longue que la durée de vie de ces sessions applicatives.

Comme le savent les lecteurs du RFC 5887, on en est loin...

Et quels sont les protocoles, les outils et les procédures qui peuvent nous aider à s'approcher de ces beaux objectifs (section 3) ? Les protocoles sont les suivants :

— Les messages RA ("*Router Advertisment*", RFC 4861) permettent d'annoncer les préfixes IPv6 utilisés sur le lien local,
— SLAAC ("*StateLess Address AutoConfiguration*", RFC 4862) permet aux machines de se configurer une adresse IPv6 après avoir entendu les RA,
— RA + SLAAC ne sont pas la seule solution, IPv6 a aussi DHCP (RFC 8415 et RFC 8415 pour la délégation de préfixe),
— Il y eu aussi un système utilisant ICMP spécifiquement pour la rénumérotation, décrit dans le RFC 2894 et qui ne semble pas avoir jamais été déployé.

Il y a aussi des outils :

— Les moyens et grands réseaux ne se gèrent pas à la main, ils se servent d'IPAM, qui gardent trace des adresses et de leur allocation,
— Dès que le réseau a une taille non triviale, il est fréquent que les modifications ne soient pas propagées manuellement mais par un outil comme Chef ou Puppet,
— Cette propagation des configurations peut aussi se faire avec des outils utilisant NETCONF (RFC 4741) ou d'autres protocoles de transport. Pour des détails pratiques, on peut consulter l'article de Leroy et Bonaventure, « "*Preparing network configurations for IPv6 renumbering*" `<http://inl.info.ucl.ac.be/system/files/dleroy-nem-2009.pdf>` »,

Et des procédures :

— Le RFC 4192 décrit en détail comment renuméroter un réseau IPv6 sans "*flag day*", sans un jour J où le soleil se coucherait sur un réseau complètement migré,
— Le RFC 6879, déjà cité, étudie des cas de renumérotation et fait des recommandations opérationnelles.

Mais, en pratique, cela ne suffit pas. La section 5 se penche sur le premier problème, la configuration des adresses dans les machines terminales et les routeurs. D'abord, l'interaction entre SLAAC et DHCP, une source de frustration fréquente pour les administrateurs réseaux IPv6. En gros, ces deux protocoles permettent de faire la renumérotation. S'ils sont présents tous les deux sur un réseau, c'est plus compliqué et il faut bien les coordonner. Le plus simple est quand même de n'en avoir qu'un seul (ce qui est, heureusement, le cas le plus courant). Normalement, il existe des indications dans les messages permettant de coordonner les deux protocoles (options M et O du RA, cf. RFC 4861, section 4.2). Mais ils sont traités comme indicatifs seulement. Résultat, les différents systèmes d'exploitation ne les interprètent pas de la même manière. On n'est donc pas sûr, par exemple, qu'on puisse renuméroter avec les RA des machines qui avaient été configurées en DHCP (`ManagedFlag` passant de 1 à 0). De même, les RFC 4862 et RFC 3315 ne sont pas clairs sur ce que doit faire une machine cliente DHCP lorsqu'elle voit des nouveaux préfixes être annoncés par des RA. Un problème analogue se pose en sens inverse (transition de DHCP vers SLAAC).

Et pour les routeurs ? Logiquement, tout devrait bien se passer sauf que, contrairement aux machines terminales qui acceptent toutes de changer d'adresse IP en vol, certains routeurs doivent être redémarrés lorsque cela leur arrive.

Comme souvent en renumérotation, on s'épargnerait bien des problèmes en utilisant des noms dans les configurations et pas des adresses IP. Mais le RFC note, à juste titre, que ce n'est pas toujours le cas pour les routeurs.

Une fois machines terminales et routeurs renumérotés, il reste à mettre à jour des tas de configurations et de bases de données qui stockent encore les anciennes adresses. C'est l'objet de la section 6. Premier exemple, les zones DNS. La zone peut être maintenue à la main, avec un éditeur et il faudra alors faire un rechercher/remplacer. Ou bien on peut utiliser les mises à jour dynamiques (RFC 3007). Cette fonction existe depuis longtemps, est mise en œuvre dans plusieurs logiciels serveurs mais, pour que chaque machine puisse individuellement faire la mise à jour, il faut un mécanisme d'autorisation, et gérer une clé dans chaque machine n'est pas pratique. Il est plus commun que la mise à jour du DNS soit faite par le serveur DHCP, pour le compte des machines terminales (RFC 4704). Cela marche très bien si la renumérotation est faite par DHCP. Si on a utilisé SLAAC, il n'existe pas de solution évidente (aujourd'hui : des réflexions sont en cours, par exemple dans `draft-ietf-dhc-addr-registration`).

Il reste à traiter le cas des adresses IP éparpillées un peu partout, notamment dans des ACL. Il n'existe pas de solution générale pour ce problème. Mais notre RFC recommande que, lorsqu'on utilise des adresses IP dans des configurations, on définisse une variable au début et on l'utilise ensuite, pour limiter la douleur de la renumérotation. Un exemple en shell Unix avec Netfilter, pour un routeur Unix qui transmet vers un serveur HTTP :

```
# Define the variable
WEBSERVER=2001:db8:cafe::1:34a

# By default, block everything
ip6tables --policy FORWARD DROP

# Allow ICMP. We could be more selective.
ip6tables --append FORWARD --protocol icmp --destination ${WEBSERVER} --jump ACCEPT

# Rate limiting
ip6tables --append FORWARD --protocol tcp --destination ${WEBSERVER} --dport 80 --tcp-flags SYN SYN -m hashlimit
   --hashlimit-name Web --hashlimit-above 3/second --hashlimit-mode srcip \
   --hashlimit-burst 7 --hashlimit-srcmask 28 --jump DROP

# Allow HTTP and HTTPS
ip6tables --append FORWARD --protocol tcp --destination ${WEBSERVER} \
         --dport 80:443 --jump ACCEPT
```

Ainsi, le script n'utilise qu'une seule fois l'adresse, au début, et est donc bien plus facile à modifier. Un programme comme ack-grep est très pratique pour trouver tous ces cas où une adresse est stockée dans la configuration :

```
# ack-grep --all 2001:db8:cafe::1:42
network/interfaces
14:        address 2001:db8:cafe::1:42

nsd3/nsd.conf
15:        ip-address: 2001:db8:cafe::1:42
...
```

Pour simplifier ce problème des adresses IP dispersées un peu partout, notre RFC recommande l'utilisation d'outils de gestion de configuration comme Chef ou Puppet : on change la configuration à un endroit et elle est ensuite poussée vers tous les serveurs. Il existe aussi des outils non-libres, spécifiques à un vendeur particulier, pour assurer cette tâche et le RFC regrette l'absence d'une norme complète (NETCONF ne règle pas tous les cas) et ouverte pour cette question de mise à jour.

Pour que l'opération de migration vers les nouvelles adresses se passe bien, il faut aussi des mécanismes de gestion des événements, notamment de notification (« Un nouveau préfixe est arrivé »). Ces notifications permettraient, par exemple, d'invalider les caches DNS (pour forcer une mise à jour), ou de changer

les configurations de filtrage (le RFC 2827 demande qu'on ne laisse pas sortir de son réseau les paquets ayant une adresse IP source autre que celles du réseau ; mettre ce filtrage en œuvre nécessite de connaître le préfixe du réseau local, et d'en changer si nécessaire). Cette notification n'est pas triviale, notamment si on utilise SLAAC (puisque, dans ce cas, aucun serveur central n'est informé du changement d'adresse d'une machine)

Le DNS pose d'ailleurs d'autres questions. En raison du TTL des enregistrements DNS, les nouvelles informations ne peuvent pas être réjuvénées <https://www.bortzmeyer.org/dns-propagation.html> instantanément. Si la rénumérotation du réseau est prévue suffisemment à l'avance, la bonne pratique est d'abaisser ces TTL avant <https://www.bortzmeyer.org/changement-adresse-et-dns.html>, de faire le changement, puis de remonter les TTL.

Les sections 9 et 10 résument les trous qu'il y a actuellement dans les protocoles TCP/IP, et dont le comblement rendrait la rénumérotation plus facile :

— Un mécanisme pour informer le routeur qu'il doit se rénuméroter lui-même, et pour choisir une adresse dans le préfixe délégué. Et une meilleure gestion de sa propre renumérotation par le routeur (ne nécessitant pas de redémarrage).
— Une meilleure spécification des interactions SLAAC¡-¿DHCP, notamment sur la renumérotation en DHCP de machines numérotées par SLAAC et réciproquement.
— Le problème de la mise à jour du DNS par les machines SLAAC reste entier.
— L'usage trop important d'adresse IP « codées en dur », non dérivées d'une unique variable facile à changer. Certains systèmes ne permettent même pas une telle dérivation.
— Un mécanisme de notification « attention, les adresses vont changer / ont changé ».
— L'interaction avec les TTL DNS reste compliquée.

La section 10 se spécialise dans les trous considérés comme insolubles, et dont le groupe 6renum ne s'occupera donc pas :

— La mise à jour des zones DNS pour le cas où elles sont extérieures à l'organisation, par exemple chez un hébergeur DNS qui fournit uniquement une interface Web pour les mettre à jour.
— La synchronisation des entrées `AAAA` (nom -¿ adresse) et `PTR` (adresse -¿ nom) dans le DNS. Cette synchronisation est d'autant plus insoluble que les zones « directes » et « inverses » (`ip6.arpa`) peuvent très bien être gérées par des hébergeurs différents.
— Créer des types d'enregistrement DNS explicitement pour faciliter la renumérotation a déjà été tenté avec le `A6` du RFC 2874. Trop complexe, le `A6` a été peu déployé et a officiellement été abandonné avec le RFC 6563. L'idée de séparer le préfixe de l'adresse et son suffixe lors de la résolution DNS reste dans l'air mais le groupe 6renum n'a pas l'intention de s'y attaquer.
— Les protocoles de transport comme TCP lient une connexion aux adresses IP de source et de destination. Cela a des tas d'inconvénients <https://www.bortzmeyer.org/separation-identificateur-
html> (en cas de renumérotation, les connexions cassent) mais c'est considéré comme un problème de grande ampleur, loin des capacités du groupe 6renum.
— Un problème analogue se pose dans **certaines** applications, qui cassent les sessions lorsqu'une adresse IP d'une des parties change.

Voilà, nous avons fait le tour des problèmes, il reste à lire la section 11, spécialisée dans les questions de sécurité. Par exemple, si on a des ACL sur les adresses IP, et qu'elles interdisent à certains méchants de se connecter à l'adresse IP d'un serveur, la rénumérotation ne doit pas invalider cette liste noire : la règle doit être mise à jour pour pointer vers la nouvelle adresse IP du serveur.

Mais on peut aussi noter que la configuration automatique, souhaitée par le RFC, amène aussi ses propres risques, comme illustré par l'accident de Cloudflare <http://seenthis.net/messages/118644> ou comme un cas rigolo avec Ansible <http://jpmens.net/2013/02/06/don-t-try-this-at-the>.