RFC 5887 : Renumbering still needs work

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 mai 2010

Date de publication du RFC : Mai 2010

https://www.bortzmeyer.org/5887.html

Changer les adresses IP utilisées par un réseau a toujours été une plaie pour l'administrateur réseaux. Il faut modifier des tas de fichiers, plusieurs configurations et on en oublie toujours. Des années après, on retrouve encore les anciennes adresses IP à des endroits inattendus (dans les documentations, par exemple). Sans compter que le changement peut mettre en jeu des partenaires extérieurs à votre organisation, par exemple parce qu'ils ont autorisé vos adresses IP dans la configuration de leur pare-feu. Bref, tout ingénieur qui a fait l'opération sait très bien qu'elle est très coûteuse.

Résultat, les gens hésitent à changer. Ils souhaitent conserver leurs adresses IP en changeant d'opérateur (les adresses PI, l'équivalent Internet de la portabilité des numéros de téléphone), ils jouent avec les règles des RIR pour garder les adresses, ils contribuent ainsi à la fragmentation de la table de routage globale.

Un certain nombre de projets de nouvelles architectures pour l'Internet, ou bien de tentatives de réduction de la taille de la DFZ repose sur l'idée que la rénumérotation d'un réseau devrait devenir banale (cf. section 1), permettant ainsi de ne garder que les noms de domaine comme identificateurs stables, les adresses IP pouvant changer sans douleur. C'était certainement l'idée originale pour la séparation entre adresses et noms <https://www.bortzmeyer.org/pourquoi-le-dns.html> mais est-ce réaliste ? Ce RFC répond qu'en tout cas, il y a du travail avant que ce ne soit le cas...

Que contient ce document ? Une analyse des mécanismes permettant ou facilitant le rénumérotation, ainsi que de leurs limites. C'est un document très concret, nourri de beaucoup d'expériences pratiques, et qui devrait intéresser tous les administrateurs réseaux. À l'IETF , espérons qu'il injectera un peu de réalisme opérationnel dans les discussions. Il succède au RFC 1900 [1], d'où son titre dérive. Plusieurs documents ont été publiés par l'IETF sur ce sujet (cf. section 1, qui cite notamment le RFC 4192). Depuis

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc1900.txt

la publication du RFC 1900 en 1996, plusieurs techniques ont été développées, qui auraient dû faire disparaître le problème (par exemple DHCP - RFC 2131 et RFC 8415 ou l'auto-configuration sans état d'IPv6 - RFC 4862). Pourtant, le diagnostic unanime des praticiens est que le renumérotage des réseaux demeure coûteux et délicat (renuméroter une machine individuelle, par exemple celle de M. Michu chez lui, alors qu'il est abonné à un FAI grand public est, par contre, en général une opération indolore).

Quelles sont les raisons pour renuméroter un réseau entier ? La section 1 fournit de nombreux cas où c'est nécessaire :

— Changement d'opérateur, pour les malheureux qui ont des adresses PA, la cause la plus courante,
— Renumérotation chez l'opérateur lui-même...
— Changement de topologie (par exemple pour tenter de récupérer quelques adresses IPv4 au fur et à mesure qu'elles s'épuisent <https://www.bortzmeyer.org/epuisement-adresses-ipv4.html>) ou d'organisation interne (par exemple fusion de deux entreprises),
— etc.

Dans tous ces cas, le changement est planifié à l'avance (la renumérotation non planifiée est bien plus difficile).

Enfin, pour terminer cette introduction, le RFC note que certaines solutions techniques, en séparant les adresses locales et les adresses publiques, peuvent diminuer et peut-être supprimer le besoin de renumérotation. C'est le cas du NAT, bien sûr, mais aussi de toutes les solutions fondées sur la séparation de l'identificateur et du localisateur <https://www.bortzmeyer.org/separation-identificateur-locali
html>. Mais le NAT a d'énormes défauts (voir par exemple le RFC 5128) et les techniques de séparation sont encore peu déployées.

Quelles sont les solutions techniques aujourd'hui pour faciliter la rénumérotation ? La section 2 explore les techniques du côté des machines non routeuses. Par exemple, DHCP (section 2.1) a été un grand succès en terme de déploiement et il rend la renumérotation d'une machine « terminale » presque indolore. Même chose pour l'auto-configuration sans état de IPv6 (sections 2.2 et 2.3) ou pour PPP (section 2.4, RFC 1332 et RFC 5072), ce dernier étant bien plus riche puisqu'il fournit des possibilités de négociation des paramètres comme les adresses.

La très grande majorité des interactions entre machines, sur l'Internet, commencent par une requête DNS. En effet, les machines et les services sont en général référencés par leur nom et pas par leur adresse IP. La rénumérotation d'un réseau va donc nécessiter la mise à jour des enregistrement DNS (section 2.5). Sur les sites sérieux, ces enregistrements sont typiquement mis à jour à partir d'une base de données, ce qui rend la renumérotation relativement simple. Une solution alternative est que la machine (ou le serveur DHCP) mette à jour le DNS (RFC 3007) Cette solution est largement déployée, et des outils existent pour toutes les plate-formes (comme `nsupdate` pour Unix).

Une limitation à l'usage du DNS pour garder des identificateurs stables (les noms de domaine) afin de pouvoir changer facilement les adresses IP est la sécurité. Celle du DNS est traditionnellement imparfaite. Au moment de la publication de notre RFC 5887, toutefois, les choses changent nettement puisque le déploiement de DNSSEC bat son plein (la racine du DNS a ainsi été signée quelques semaines avant la publication du RFC).

Encore un cran au dessus de l'utilisation des noms de domaine se trouve la découverte de services dans le DNS. C'est ainsi que SLP (RFC 2608) ou bien des solutions non-standard comme Bonjour permettent de découvrir, par exemple, l'imprimante du réseau local.

Cette section 2 était consacrée aux mécanismes utilisés par les machines non-routeuses. Et sur les routeurs ? La section 3 les étudie. Par exemple, l'option "*Prefix Delegation*" de DHCP (RFC 3633) permet

d'indiquer à un routeur le préfixe IPv6 qu'il va devoir router (section 3.1). Il existe aussi un mécanisme ICMP, normalisé dans le RFC 2894, mais qui semble n'avoir jamais été déployé. Et le RFC 4191 fournit également un service intéressant d'enrichissement des "*Router Advertisements*" d'IPv6.

Le cas spécifique d'IPv6 est traité également en section 4, qui note que, contrairement à son prédécesseur, IPv6 a été prévu dès le début pour une coexistence de préfixes d'adresses sur le même réseau. Cela permet théoriquement des renumérotations plus faciles, en installant le nouveau préfixe sans supprimer l'ancien immédiatement (cf. RFC 4192). En outre, le mécanisme des ULA (RFC 4193) permet d'avoir des adresses locales uniques (ce qui n'est pas possible avec les adresses privées IPv4 du RFC 1918). D'autre part, l'existence des adresses IP temporaires du RFC 8981 fait que les mises en œuvres d'IPv6 sont normalement préparées à des fréquents changements d'adresse, ce qui devrait, en théorie, faciliter les rénumérotations.

Avec tous ces mécanismes, comment se présente en pratique la renumérotation d'un réseau ? La section 5 descend dans le concret en étudiant ce point. Du côté des machines non-routeuses, une première série de problèmes concerne la couche 3 (section 5.1.1). La grande majorité de ces machines obtient son adresse par DHCP et la garde pour la durée du bail. L'administrateur réseau compétent, qui planifie à l'avance, peut donc abaisser la durée du bail lorsque le moment de la renumérotation approche, changer le préfixe, puis remonter la durée du bail. On peut ainsi renuméroter uniquement avec DHCP. Il existe aussi une extension DHCP IPv4, `FORCERENEW` (RFC 3203) pour forcer une mise à jour immédiate, si on a oublié de planifier. Mais elle semble peu disponible en pratique.

DHCP n'est pas qu'une aide, il peut aussi être lui-même une source de problèmes. Il a actuellement 170 options (!) et certaines transportent des adresses IP, et la configuration de ces options doit être changée en cas de renumérotation.

L'auto-configuration sans état d'IPv6 (SLAAC - "*StateLess Address AutoConfiguration*") permet également une renumérotation facile, d'autant plus qu'elle a moins d'options. Par contre, si on utilise SLAAC et DHCP en même temps (RFC 8415), il faut prendre garde à leur interaction, qui n'est pas normalisée avec suffisamment de précision.

DHCP et SLAAC n'aident pas si les adresses sont marquées en dur dans la machine (par exemple pour éviter de dépendre du serveur DHCP), ou bien si la machine n'a pas de client DHCP. Cela arrive dans le monde de l'embarqué où l'adresse IP doit parfois se configurer par des commutateurs DIP.

DHCP (ou bien SLAAC) permettent de changer facilement l'adresse IP. Mais cela peut perturber les autres couches. Par exemple, TCP identifie une connexion par un tuple qui comprend entre autres les adresses IP source et destination (et la somme de contrôle les utilise donc on ne peut pas facilement les changer dans le dos de TCP). Renuméroter signifie qu'on casse les connexions TCP existantes (section 5.1.2). Ce n'est pas forcément un problème pour HTTP, où les connexions sont en général courtes, mais c'est bien plus gênant pour SSH. D'autres protocoles de transport permettent par contre la renumérotation (SCTP, RFC 4960) mais ils sont peu déployés.

Continuant à grimper vers les couches hautes, le RFC note que le DNS nécessite une attention particulière (section 5.1.3). Si les données servies par le DNS sont produites à partir d'une base de données centrale, l'opération de renumérotation est relativement simple. Il faut toutefois penser à abaisser le TTL à l'avance, pour éviter, une fois le changement fait, que les vieilles informations traînent dans les caches.

Si, par contre, le DNS est géré à l'ancienne, avec des fichiers de zone édités à la main, le DNS devient alors un problème de plus en cas de renumérotation. Si le DNS, comme c'est souvent le cas, est sous-traité, il y aura en plus une étape de coordination avec l'hébergeur.

Ensuite, il y a les problèmes liés aux applications proprement dites, la couche 7 (section 5.1.4). Tous les protocoles applicatifs n'ont pas les mêmes problèmes. Le RFC 3795 avait étudié 257 protocoles IETF et découvert que 34 stockaient explicitement les adresses IP, ce qui les rend peu utilisables en cas de renumérotation. (Le plus connu est FTP, qui passe l'adresse IP dans la commande `PORT` , section 4.1.2 du RFC 969.) Mais l'analyse des protocoles ne suffit pas, encore faut-il étudier le comportement des applications elle-mêmes. Celles-ci stockent souvent des adresses IP, sans penser qu'elles puissent changer.

Ainsi, beaucoup d'applications font du "*pinning*" : elles résolvent le nom en adresse IP une seule fois et ne modifient plus cette information, même si l'adresse IP a changé. À leur décharge, il faut préciser que la routine standard `getaddrinfo()` ne fournit pas d'information de durée de vie... (Le RFC suggère aux applications de respecter le TTL du DNS en oubliant que `getaddrinfo` ne le transmet pas. Suivre ce conseil nécessiterait donc que les applications fassent leur propres requêtes DNS.) Idéalement, l'application devrait vérifier la validité de l'adresse (par exemple en refaisant `getaddrinfo()` à chaque ouverture de connexion avec une machine distante).

Les applications les plus sensibles sont celles qui restent ouvertes longtemps, comme les navigateurs Web. Ceux-ci, en général, sont conscients du problème et ne gardent l'adresse IP que pendant un temps limité mais ils épinglent parfois délibérement cette adresse, pour des raisons de sécurité Javascript <https://www.bortzmeyer.org/dns-rebinding-pinning.html> (cf. section 8). D'autres applications traitent le problème (ainsi que celui, plus général, d'une connectivité imparfaite) en essayant régulièrement et sur toutes les adresses possibles (les applications pair-à-pair font souvent cela). C'est non-trivial à développer.

Les dépendances des applications vis-à-vis des adresses IP sont souvent pires : par exemple, il existe des mécanismes de licence logicielle où la licence dépend de l'adresse IP... Plus légitimes, il y aussi les "*cookies*" HTTP liés à une adresse IP. (L'annexe A donne d'autres exemples.)

Mais le RFC note aussi que, par delà la limite de `getaddrinfo()` citée plus haut (pas d'information sur la durée de vie d'une adresse), le problème des API est plus général : celles-ci sont en général de trop bas niveau, obligeant les applications à stocker des adresses IP ce qui, en général, ne devrait pas être nécessaire <https://www.bortzmeyer.org/network-high-level-programming.html>. La traditionnelle API "*socket*", conçue avant même le DNS, est ainsi en tort. Dans le futur, il faut espérer que les programmes utiliseront des API de plus haut niveau, n'exposant pas du tout les adresses IP. C'est typiquement le cas en Java et, en C, il existe des bibliothèques pour cela comme libcurl.

Après les machines non-routeuses, place aux routeurs (section 5.2). Depuis le RFC 2072, où en sommes-nous ? Il y a eu des progrès mais des points soulevés par ce vieux RFC sont toujours d'actualité. Par exemple, pour la configuration des tunnels, bien que l'utilisation d'un nom de domaine pour configurer IPsec soit normalisée (RFC 4306, section 3.5), elle reste peu utilisée. La configuration du routeur contient ainsi les adresses IP des extrémités des tunnels, gênant ainsi la renumérotation.

La section 5.3 parcourt d'autres questions qui ne sont pas liées aux routeurs ou aux machines terminales. Par exemple, la section 5.3.4 est un excellent et très concret examen des problèmes d'administration système. La situation idéale serait que l'information sur les machines, leurs connexions et leurs adresses soient dans une base de données centrale et que tous les fichiers de configuration soient fabriqués automatiquement à partir de cette base. C'est loin d'être la réalité partout et, en pratique, la renumérotation nécessite des grands coups de `grep` dans `/etc` (voire ailleurs) pour trouver toutes les occurrences des vieilles adresses IP avant de les changer. Faites l'essai sur votre site : dans combien de fichiers sont stockées vos adresses ? Pour les règles du pare-feu, ou le DNS, vous savez sans doute les trouver. Mais il y a aussi des adresses IP à plein d'endroits surprenants, parfois chez des tiers.

Pourquoi les fichiers de configuration utilisent-ils si souvent des adresses IP lorsque des noms de domaine conviendraient ? Le RFC 1958, section 4.1, disait déjà en 1996 que c'était une mauvaise idée. C'est souvent le pur conservatisme qui empêche d'adopter cette bonne pratique. Mais il y a plusieurs bonnes raisons, l'une d'elles étant que la traduction de nom en adresse IP ne se fait souvent qu'une fois (par exemple, pour un routeur, au démarrage) et qu'il faudra donc de toute façon au moins redémarrer en cas de renumérotation. Une autre raison (RFC 1958, section 3.11), est pratique : si la configuration du routeur utilise des noms, il dépendra du DNS qui lui-même dépend du routage, créant ainsi une dépendance circulaire, source de problèmes.

Autre raison pour laquelle beaucoup d'administrateurs utilisent des adresses et pas des noms : la sécurité (section 5.3.5). Le DNS n'étant pas sûr (cf. RFC 3833), configurer un outil de sécurité, comme le pare-feu, via des noms n'est pas logique. Le déploiement de DNSSEC, plutôt rapide en ce moment, pourrait résoudre cette question.

Il reste bien des questions de sécurité liées à la rénumérotation, dans cette section 5.3.5 (voir aussi la section 8), comme le risque d'annonces de fausses adresses (presque personne n'utilise le RFC 3971) pendant l'opération, le risque lié à un pare-feu mis à jour trop tôt (et il bloquerait les anciennes adresses à tort) ou trop tard (bloquant les nouvelles adresses), les certificats X.509 contenant une adresse IP (RFC 5280)...

Quels sont les mécanismes nouveaux qui sont proposés pour faciliter les futures renumérotations, se demande la section 6 :

— SHIM6 (RFC 5533) est un mécanisme de "*multi-homing*" (avoir plusieurs adresses et les gérer correctement) situé sur la machine non-routeuse, comme HIP, SCTP ou d'autres,
— Le groupe de travail MANET `<http://tools.ietf.org/wg/manet>` travaille à des solutions de mobilité pour les réseaux, ce qui résoudrait en même temps la question de la rénumérotation,
— Des protocoles comme Netconf (RFC 6241) pourraient faciliter les choses, en permettant l'administration centrale et standard des équipements réseau.

Et quels sont les « trous », les points qui manquent pour faire de la renumérotation un évenement banal et simple ? La section 7 les étudie successivement.

Il serait souhaitable, comme indiqué plus haut, que l'API indique la durée de vie d'une adresse (section 7.1). Une interface qui ne manipulerait que des noms `<https://www.bortzmeyer.org/programmation-orientee-nom.html>` serait encore meilleure.

Un moyen standard de stocker et de récupérer la configuration des machines serait également une grande aide (il existe déjà plein de moyens ad hoc et non standards). Cela concerne aussi bien les routeurs que les machines ordinaires, et cela pourrait utiliser Netconf.

Et peut-être pourrait-on étendre UDP et TCP de manière à les rendre multi-adresses (comme l'est SCTP).

Pour les routeurs (section 7.2), déployer enfin les RFC 2894 et RFC 3633 aiderait beaucoup.

Enfin, puisque ce RFC prend soin de ne pas oublier les considérations opérationnelles, la section 7.3 identifie les manques actuels dans ce domaine. D'abord, comme déjà cité, tant que DNSSEC n'est pas largement déployé, il sera difficile d'exiger des administrateurs système qu'ils renoncent à l'usage des adresses IP dans les fichiers de configuration.

Ensuite, nous devons pousser au déploiement de techniques d'administration système plus modernes, dépendant moins de l'édition manuelle de dizaines de fichiers avec vi.

Finalement, il reste à documenter et à écrire de jolis HOWTO sur la renumérotation d'un réseau... (Un exemple est « "*Preparing network configurations for IPv6 renumbering*" <http://inl.info.ucl.ac.be/system/files/dleroy-nem-2009.pdf> ».)

Quelques conseils pratiques personnels, tirés de l'expérience de renumérotations réussies ou ratées :

— Planifiez bien à l'avance, en écrivant un plan et un calendrier. Une maîtrise des aspects temporels est particulièrement nécessaire avec des protocoles comme le DNS, où la vieille information peut être gardée dans les caches.
— Il peut être intéressant d'utiliser un système d'administration centralisé comme cfengine ou puppet,
— Pour génerer des fichiers de configuration, si on n'a pas confiance dans le DNS et donc dans les noms, si on hésite à déployer un système complexe avec base de données centrale et tout le toutim, une solution simple et légère est simplement de passer par un préprocesseur, comme cpp ou m4. On peut alors définir la correspondance nom¡-¿adresse au début du script (voire dans un fichier central) et utiliser ensuite des noms dans la configuration.

Sur beaucoup de sites, simplement déterminer tous les endroits où l'adresse IP est marquée peut être difficile. grep aide évidemment beaucoup mais, pour des recherches dans toute une arborescence, je préfère ack-grep, par exemple, avec l'adresse IPv4 actuelle de la machine qui héberge ce blog :

```
% sudo ack --all 208\.75\.84\.80
apache2/vhosts.d/drupal.example.net.conf
74:     Allow from 208.75.84.80

conf.d/net
3:config_eth0=( "208.75.84.80 netmask 255.255.255.0 broadcast 208.75.84.255" )

nsd/nsd.conf
17:        ip-address: 208.75.84.80
```

Pas mal : une renumérotation ne devrait pas être trop dure. Vous noterez que l'adresse IP n'apparait pas dans la configuration du pare-feu : Shorewall <https://www.bortzmeyer.org/filtrage-avec-shorewall.html> la trouve tout seul, supprimant un problème de mise à jour.

En conclusion, une analyse personnelle : l'état de la technique aujourd'hui, avec les grandes difficultés de renumérotation qui existent, justifie amplement l'existence des adresses PI, même si les gros opérateurs, dans leurs forums comme le RIPE-NCC, les regardent souvent comme un luxe. Comme renuméroter est difficile, s'il n'existait que des adresses PA, les utilisateurs auraient beaucoup de mal à changer de fournisseur...