RFC 6180 : Guidelines for Using IPv6 Transition Mechanisms during IPv6 Deployment

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 mai 2011

Date de publication du RFC : Mai 2011

https://www.bortzmeyer.org/6180.html

IPv6 fait souvent peur aux administrateurs réseaux par le nombre de techniques de coexistence entre IPv4 et IPv6. Ces techniques sont variées, et inévitables, puisque la transition a pris un retard considérable <https://www.bortzmeyer.org/ipv6-et-l-echec-du-marche.html>, ce qui mène à devoir déployer IPv6 alors que les adresses IPv4 sont déjà épuisées <https://www.bortzmeyer.org/epuisement-adresses-ipv4.html>. Ce RFC 6180 [1] vise à simplifier la tâche dudit administrateur système en le guidant parmi les techniques existantes.

Il ne s'agit donc pas de normaliser le Nième protocole de transition et/ou de coexistence entre IPv6 et IPv4, plutôt de faire le point sur les protocoles existants, dont le nombre a de quoi effrayer. Mais il ne faut pas paniquer : selon le cas dans lequel on se situe, toutes ces techniques ne sont pas forcément pertinentes et il n'est donc pas nécessaire de les maîtriser toutes.

D'abord, notre RFC 6180 rappelle qu'il n'y a pas le choix : la réserve IPv4 de l'IANA a été épuisée en février 2011 et celle des RIR le sera fin 2011-début 2012. IPv6 est donc la seule solution réaliste. Le point sur la transition est fait dans le RFC 5211 et elle a commencé il y a longtemps, dans les organisations les mieux gérées. Un des points qui la freine est que l'administrateur réseaux normal, celui qui ne suit pas heure par heure les travaux de l'IETF, voit un grand nombre de techniques de coexistence IPv4/IPv6 (le Wikipédia anglophone a excellente page de synthèse sur ces techniques) et se demande « Dois-je donc déployer 6to4 **et** 6rd **et** DS-lite **et** NAT64 et donc maîtriser toutes ces techniques ? ». La réponse est non : selon son réseau, notre malheureux administrateur n'aura à gérer **qu'une partie** de ces méthodes.

Déjà, il faut comprendre ce que ces termes barbares veulent dire : la section 2 rappelle la terminologie, et les RFC à lire.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6180.txt

La section 3 décrit ensuite les principes suivis dans le RFC : le premier est de garder en tête le but, qui est de pouvoir continuer la croissance de l'Internet sans être gêné par des problèmes comme le manque d'adresses. Pour cela, le but final de la transition doit être le déploiement complet d'IPv6 natif dans tout l'Internet. Lorsque cela sera réalisé, toutes les techniques de transition et de coexistence pourront être abandonnées, simplifiant ainsi beaucoup la situation. Mais pour atteindre ce but lointain, il n'y a pas qu'une seule bonne méthode. Par contre, ces méthodes ont quelques points en commun. Il est par exemple essentiel de s'y prendre à l'avance : les adresses IPv4 sont déjà épuisées et une organisation qui n'aurait encore rien fait du tout, question IPv6, va avoir des problèmes. Donc, aujourd'hui, En tout cas, il n'existe pas de raison valable de retarder le déploiement d'IPv6. Ceux qui n'ont pas encore commencé devraient s'y mettre immédiatement.

Mais cela implique une longue période de coexistence avec IPv4. Pas pour tout le monde : il existe des réseaux un peu particuliers, des déploiements récents, contrôlés par une seule organisation, dans un environnement fermé (une usine, par exemple) et qui n'ont pas besoin d'échanger avec le reste du monde. Ceux-ci peuvent faire de l'IPv6 pur et ne pas lire ce RFC. Mais tous les autres doivent se poser la question « Qu'est-ce que je fais avec mes deux protocoles ? ».

Et cela va durer longtemps. Même si on peut espérer qu'un jour on éteigne la dernière machine IPv4, cela ne se produira pas avant de nombreuses années. Même si un réseau est entièrement IPv6, il faudra bien qu'il puisse communiquer avec des machines v4 à l'extérieur. (Le RFC ne mentionne pas le fait qu'actuellement, l'effort de la coexistence repose entièrement sur ceux qui veulent déployer IPv6. Une date importante sera celle où l'effort changera de camp et où ce sera ceux qui veulent continuer à utiliser IPv4 qui devront travailler pour cela. Cette date surviendra longtemps avant la disparition de la dernière machine IPv4.)

Ceux qui déploient IPv6 doivent donc choisir un modèle de déploiement. Le RFC 5218 rappelait, en s'inspirant de l'expérience de plusieurs protocoles, ce qui marche et ce qui ne marche pas. Certains de ces rappels semblent évidents (ne pas casser ce qui marche aujourd'hui, permettre un déploiement incrémental, puisqu'il n'y a aucune chance de faire basculer tout l'Internet d'un coup) mais sont toujours bons à garder en mémoire.

Voilà, après ces préliminaires, place aux techniques elle-mêmes, en section 4. Le début de cette section précise qu'elle n'est pas exhaustive : il existe d'autres mécanismes de coexistence/transition mais ils sont considérés comme relativement marginaux. Les techniques sont exposées dans l'ordre d'importance décroissante et les premières sont donc les plus recommandées.

Premier mécanisme de coexistence, et le plus recommandé, la **double pile**. Chaque machine a deux mises en œuvre des protocoles réseau, une pour IPv4 et une pour IPv6 et peut parler ces deux protocoles au choix. Ce modèle s'applique aussi bien sur la machine terminale (où les applications sont donc « bilingues ») que sur les routeurs du FAI (qui font tourner des protocoles de routage pour les deux familles, cf. RFC 6036 pour le point de vue du FAI). C'était le mécanisme de transition originellement envisagé. Le choix de la version d'IP dépend de la machine qui initie la connexion (RFC 6724), et de ce que la machine réceptrice a annoncé (par exemple dans le DNS). Ce mécanisme est simple et évite les problèmes qu'ont, par exemple, les tunnels, comme les problèmes de MTU. Comme le rappelle le RFC 4213, c'est la méthode recommandée.

Cela n'empêche pas ce modèle d'avoir des failles. D'abord, si l'adoption de ce modèle est une décision individuelle de chaque réseau, son utilisation nécessite que les deux réseaux qui communiquent aient adopté ce système. Si un client double-pile veut se connecter à un serveur en IPv6, celui-ci doit avoir une connectivité IPv6 (ce qui dépend de son FAI), doit avoir une application IPv6isée et doit avoir annoncé un AAAA dans le DNS. Autrement, on a IPv6, mais on n'observe guère de trafic. Inversement,

lorsqu'un gros fournisseur de contenu active IPv6 (comme l'a fait YouTube en février 2010), le trafic IPv6 fait soudain un bond.

Une deuxième limite de l'approche double-pile est que certaines applications ne réagissent pas proprement lorsqu'un des protocoles fonctionne mais pas l'autre. De nos jours, c'est en général l'IPv6 qui a un problème et la double-pile peut alors entraîner une dégradation du service, l'application perdant bêtement du temps à tenter de joindre le pair en IPv6 au lieu de basculer tout de suite vers IPv4. Le code naïf de connexion d'un client vers un serveur est en effet (en pseudo-code) :

```
for Address in Addresses loop
    try
       connect_to(Address)
       exit loop
    except Timeout
       # Try the next one
    end try
end loop
```

Cette approche purement séquentielle peut être pénible si les adresses IPv6 sont en tête de la liste `Addresses` et si le délai de garde est long. Car chaque connexion fera patienter l'utilisateur plusieurs secondes. (La durée exacte avant le `Timeout` dépend de la nature de la non-connectivité IPv6 - le routeur envoie-t-il un message ICMP ou pas, de ce que transmet la couche 4 à l'application, etc.) C'est pour cette raison que de nombreux gérants de gros services Internet hésitent à activer IPv6 (par exemple, ce dernier est disponible pour le service, mais n'est pas annoncé dans le DNS), de peur de pourrir la vie d'une partie, même faible, de leurs utilisateurs. C'est ainsi que `www.google.com` n'a d'enregistrement AAAA que pour certains réseaux, que Google estime suffisamment fiables en IPv6. Ce mécanisme de « liste blanche » où Google décide qui va pouvoir se connecter à eux en IPv6 est contestable mais n'oublions pas que les autres « gros » sites Internet ne font rien du tout.

Rendre les applications plus robustes vis-à-vis de ce problème pourrait aider (voir par exemple la technique proposée par l'ISC <`http://www.isc.org/community/blog/201101/how-to-connect-to-a-multi-`
ou bien le très bon guide de programmation IPv6 d'Étienne Duble <`https://edms.cern.ch/document/971407`>), surtout si ces mécanisme de connexion intelligents (tenter en parallèle les connexions v4 et v6) sont emballés dans des bibliothèques standard. Par exemple, la plupart des applications pair-à-pair gèrent relativement bien ce problème, habituées qu'elles sont à vivre dans un monde de connectivité incertaine et intermittente.

Ces défauts sont à garder en tête mais il n'en demeure pas moins que la double-pile reste la meilleure méthode, d'autant plus que de nombreux réseaux ont aujourd'hui IPv6 (le RFC cite les NREN comme Renater ou Internet2, complètement IPv6 depuis longtemps).

Une conséquence amusante de la double-pile est qu'elle nécessite une adresse IPv4 par machine. Cela ne semblait pas un problème au moment où ce mécanisme a été conçu, puisqu'il semblait possible de passer tout le monde en IPv6 avant l'épuisement des adresses v4. Comme cela ne s'est pas fait, on se retrouve aujourd'hui dans une situation où il n'y a plus d'adresses IPv4. La double pile est donc en général utilisée avec une adresse IPv6 publique et une IPv4 privée, NATée plus loin.

La double pile, c'est très bien mais que faire si on n'a pas de connectivité IPv6, et qu'on veut relier son réseau IPv6 au reste de l'Internet v6 ? La section 4.2 expose la solution des tunnels. L'intérêt des tunnels est qu'ils ne nécessitent aucune action sur les routeurs situés sur le trajet. On n'a pas à attendre que son FAI se bouge le postérieur et déploie IPv6. C'est une technique connue et éprouvée, qui est d'ailleurs utilisée pour bien d'autres choses qu'IPv6. L'inconvénient des tunnels est la complexité supplémentaire,

et le fait qu'ils réduisent la MTU entraînant tout un tas de problèmes avec les sites qui bloquent stupidement l'ICMP.

Il existe plusieurs types de tunnels : les tunnels manuels du RFC 4213, des tunnels automatiques comme 6to4 (RFC 3056), les serveurs de tunnel présentés dans les RFC 3053 et RFC 5572, et bien d'autres, résumés dans le RFC 5565.

Lorsque le tunnel fait partie d'une solution gérée par un administrateur réseaux compétent, il ne pose pas de problème en soi. Mais certaines solutions techniques (comme 6to4) sont prévues pour être « non gérées » et elles ont toujours posé beaucoup de problèmes. Ainsi, elles peuvent donner aux applications l'impression qu'une connectivité IPv6 fonctionne alors qu'en fait les paquets ne reviennent pas. Beaucoup d'applications tentent alors de se connecter en IPv6 et mettent longtemps avant de s'apercevoir que cela ne marchera pas et qu'il vaut mieux se rabattre sur IPv4. 6to4 a donc beaucoup contribué à la réputation d'IPv6 comme technologie fragile et source d'ennuis. (La solution 6rd - RFC 5969, quoique dérivée de 6to4, n'a pas ces défauts.)

Pour l'instant, IPv6 est peu déployé et ce sont les gens qui veulent utiliser ce protocole qui doivent faire des efforts pour se connecter. Désormais que les adresses IPv4 sont épuisées, on commencera bientôt à voir des déploiements purement IPv6 et il faudra alors faire des efforts pour maintenir les vieux systèmes v4. Les sections 4.3 et 4.4 couvrent ces cas. En 4.3, la question est celle d'un nouvel entrant dans le monde des opérateurs qui n'a pas eu d'adresses IPv4 pour son beau réseau tout neuf et qui a donc déployé un cœur purement v6 ce qui, après tout, simplifie sa configuration. Mais ses clients, et les partenaires auxquels ses clients essaient de se connecter, sont restés en v4. Comment leur donner la connectivité qu'ils désirent ? Il va falloir cette fois tunneler IPv4 sur IPv6. Le modèle recommandé est DS-lite ("*Dual Stack Lite*"), dont le RFC n'est pas encore publié. Le principe est que le client recevra uniquement des adresses IPv4 privées, que le CPE fourni par le fournisseur d'accès encapsule les paquets IPv4 qu'émettrait un client, les tunnele jusqu'à un équipement CGN qui décapsulera, puis procédera au NAT44 classique. Le nouveau FAI aura donc quand même besoin de quelques adresses IPv4 publiques. Il existe déjà une mise en œuvre en logiciel libre de la fonction CGN, AFTR <http://www.isc.org/software/aftr>.

Autre cas, légèrement différent, est celui où le réseau local connecté à l'Internet n'a pas d'équipement IPv4 du tout. Tout beau, tout neuf, toutes ses machines (et les applications qu'elles portent) sont purement IPv6. Ce n'est pas aujourd'hui très réaliste avec des machines et des applications ordinaires mais cela pourrait arriver avec de nouveaux déploiements dans des environnements modernes. Le problème, décrit en section 4.4, est alors de se connecter quand même à d'éventuels partenaires restés v4. Une des solutions est le passage par un relais (j'en ai fait l'expérience lors d'un atelier de formation et Apache fait un très bon relais pour HTTP, permettant aux machines purement IPv6 de voir tout le ouèbe v4). Une autre solution est NAT64 (RFC 6144).

Puisqu'un certain nombre de trolls ont deversé du FUD sur la sécurité d'IPv6, la section 7, consacrée à ce sujet, est une bonne lecture. Elle rappelle que IPv6 a en gros le même niveau de sécurité qu'IPv4, c'est-à-dire pas grand'chose et que ce sont les implémentations et les déploiements qui apportent des risques, plus que les spécifications. Par contre, chaque technique de transition a ses propres risques de sécurité, en plus de ceux d'IPv6 ou d'IPv4 mais notre RFC ne les détaille pas, renvoyant aux normes décrivant ces techniques.

Conclusion ? La section 5, après cette longue énumération de bricolages variés recentre le débat : l'essentiel est d'activer IPv6. Une fois ce principe posé, cette section rappelle des points mentionnés au début comme le fait qu'il ne faut pas appliquer aveuglément la même technique à tous les réseaux. Ainsi, un réseau tout neuf ne fera sans doute pas les mêmes choix qu'un réseau existant depuis vingt et ayant accumulé plein de technologies historiques.

La section 6 propose une bibliographie pour ceux qui veulent approfondir le sujet : plein de RFC dont les RFC 4213, RFC 4038, RFC 6036, etc. D'autre part, j'avais fait un exposé au GUILDE à Grenoble sur ces sujets (IPv6 et la transition), exposé dont les transparents sont disponibles <https://www.bortzmeyer.org/transition-ipv6-guilde.html>.