RFC 5505 : Principles of Internet Host Configuration

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 20 mai 2009

Date de publication du RFC : Mai 2009

https://www.bortzmeyer.org/5505.html

Comment est-ce qu'une machine arrive à se connecter à Internet alors que tant de paramètres doivent être configurés, comme l'adresse IP, la longueur du préfixe de celle-ci, les résolveurs DNS à utiliser, etc ? Ces réglages qui se faisaient autrefois à la main sont maintenant souvent automatisés. Ce RFC de l'IAB explique les principes de la configuration IP et les meilleures façons de la faire.

Commençons par la section 1.2 qui explique les paramètres en jeu. Ils sont séparés en deux parties, les paramètres de couche 3 (section 1.2.1) et ceux des couches supérieures (section 1.2.2). Parmi les paramètres de la couche réseau, il y a bien sûr l'adresse IP, dont dépend tout le reste. Souvent configurée autrefois avec le protocole RARP, elle est désormais souvent fixée par DHCP sur IPv4 et par DHCPv6 et l'auto-configuration sur IPv6. Ces protocoles ont en effet en commun de fonctionner entièrement sur IP, contrairement à RARP. La configuration d'IP est donc auto-suffisante.

Les autres paramètres sont souvent configurables par DHCP, le routeur par défaut, le fichier à charger si la machine charge son système d'exploitation via le réseau, la MTU du lien, etc.

Bien sûr, on peut toujours configurer ces paramètres manuellement, ce qui est souvent fait pour des serveurs, qui changent rarement de paramètres, et n'aiment pas dépendre de mécanismes externes comme DHCP. Par exemple, sur une machine Debian, on configurera ces paramètres dans le fichier `/etc/network/interfaces` :

```
# Configurer l'interface eth0 avec une adresse fixe
iface eth0 inet static
        # Adresse IPv4
address 80.67.170.53
        # Longueur du préfixe, ici 26 bits, sur les 32 d'une adresse v4
netmask 255.255.255.192
        # Routeur par défaut
gateway 80.67.170.1
        ...
```

Les paramètres des couches supérieures incluent les adresses des serveurs de noms (DNS la plupart du temps mais il y a aussi des mécanismes de résolution de noms comme LLMNR qui ne nécessitent pas de configuration), service de synchronisation d'horloge (typiquement NTP), etc.

Quels sont les bons principes pour la configuration de ces paramètres ? La section 2 tente de répondre à ces questions. Premier principe, posé en section 2.1 : « Tout ce qui peut être configuré peut être configuré de travers ». Notre RFC rappelle donc le RFC 1958 [1] qui disait « Évitez les options et les paramètres autant que possible. Les options et les paramètres devraient être configurés ou négociés dynamiquement, plutôt qu'à la main ». L'idée est qu'on éviterait bien des erreurs en découvrant les paramètres automatiquement plutôt qu'en demandant à l'administrateur système de les fixer. Par exemple, la MTU peut être découverte par les techniques du RFC 4821. Tout protocole devrait donc avoir le moins de réglages possible ou, en tout cas, ceux-ci devraient pouvoir être obtenus automatiquement.

En outre, le processus de configuration devrait, même s'il est automatique, être **simple**. C'est ce que détaille la section 2.2 qui cite encore le RFC 1958 : « Que cela reste simple. En cas de doute lors de la conception d'un protocole, choisir la solution la plus simple. » Une des raisons de ce choix est qu'IP tourne sur un très grand nombre de machines, du téléphone portable au super-calculateur et ceux-ci ont des capacités très variables. Le code IP doit donc pouvoir tenir dans une machine de bas de gamme. Un exemple typique concerne les machines qui doivent charger le noyau de leur système d'exploitation depuis le réseau (PXE). Puisque, par définition, le système n'est pas en route lors de ce chargement, le code qui l'effectue doit pouvoir tenir dans une ROM et donc être très petit (cela exclut TCP, par exemple, et c'est une raison de plus de n'embarquer qu'IP, sans protocoles de configuration additionnels).

Et le choix des mécanismes de configuration ? La section 2.3 demande de le minimiser : un vaste ensemble de tels mécanismes augmenterait la taille du code, le temps de configuration (il faudrait essayer successivement plusieurs méthodes) et l'interopérabilité (au cas où deux machines du même réseau ne mettent pas en œuvre les mêmes mécanismes).

Moins évidente est la nécessité de ne **pas** dépendre des couches basses (typiquement la couche 2) pour la configuration. Ce point fait l'objet de la section 2.4. Une implémentation d'IP va typiquement tourner sur plusieurs types de réseaux et ne peut pas compter sur une technique qui serait disponible sur tous. La méthode recommandée est donc qu'IP se débrouille seul.

Il existe des contre-exemples dans la famille IP. Avec PPP, le protocole IPv4CP du RFC 1332 assurait la configuration IP et une extension à PPP (RFC 1877) configurait les serveurs de noms. Cette erreur (compréhensible à l'époque, où DHCP était très rare) a été corrigée dans son équivalent IPv6, IPv6CP (RFC 5072, mais il y a aussi d'autres mécanismes de configuration en IPv6). Le RFC 3818 a ainsi sérieusement limité la possibilité de rajouter des mécanismes de configuration dans PPP.

En outre, la couche 2, en croyant bien faire, peut en fait limiter les possibilités des machines IP. Ainsi, si l'adresse IPv6 était configurée par la couche 2, des services comme les adresses temporaires de protection de la vie privée (RFC 4941) ou comme les adresses cryptographiques du RFC 3972 ne pourraient pas être utilisées.

Enfin, la section 2 se clôt sur un point important : la configuration n'est pas le contrôle d'accès et devrait en être complètement séparée (section 2.5). De toute façon, l'administrateur d'une machine peut toujours la configurer à la main et vouloir limiter l'accès au mécanisme de configuration n'a donc guère

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc1958.txt`

de sens. Si on veut limiter l'accès au lien de niveau 2, il vaut mieux utiliser une technique spécifique telle que 802.1X.

Une fois ces bons principes posés, le RFC discute dans la section 3 quelques questions supplémentaires. 3.1 estime que l'ensemble des mécanismes de configuration existants suffit et qu'il n'est sans doute pas opportun d'en développer d'autres. Cette section rappelle également que l'Internet n'est pas qu'une affaire de machines mais aussi d'humains qui les gèrent et que les considérations organisationnelles doivent être prises en compte lors du choix d'un mécanisme de configuration. Ainsi, routeurs et serveur de noms sont en général sous la responsabilité de deux équipes différentes et qu'il faut veiller à ce que la configuration du routeur ne dépende pas inutilement d'une autre équipe.

La frontière est floue entre configuration et « découverte d'un service » comme le fait SLP (RFC 2608). La section 3.2 explore cette frontière et explique que les deux ont leur place, les mécanismes de découverte de service ayant des capacités plus étendues.

Cela mène le RFC à la question du « destin commun » ("*fate sharing*", section 3.2.1). On parle de destin commun quand deux services ont intérêt à être liés (la panne de l'un entraîne celle de l'autre) car ils n'ont pas de sens séparément. Ainsi, apprendre le routeur par défaut par des RA ("*router advertisement*", RFC 4861) émis par le routeur lui-même est un excellent exemple de destin commun. Si le routeur est en panne, il n'émettra plus de RA et les machines n'apprendront donc pas son adresse. En revanche, si le routeur est appris par DHCP, la panne du routeur ne sera pas forcément détectée par le serveur DHCP et les machines apprendront l'adresse d'un routeur inutilisable.

Le destin commun est donc en général souhaitable pour les protocoles de configuration mais il n'est pas toujours possible.

La section 4 est consacrée à la sécurité, problème toujours difficile d'autant plus qu'une machine pas encore configurée, par définition, ne sait pas encore grand'chose du réseau auquel elle est attachée et est donc très vulnérable à toutes les impostures. Si l'attaquant réussit à convaincre la machine qui se configure d'utiliser tel serveur de noms, que l'attaquant contrôle, alors presque toute l'activité Internet de cette machine est dirigée par l'attaquant. C'est encore pire si l'attaquant convainc une machine sans disque local de charger comme système d'exploitation un programme choisi par l'attaquant.

La section 4.1 explore les solutions possibles. Disons tout de suite qu'il n'y a pas de miracle : la simplicité et la souplesse de la configuration d'IP s'opposent à la sécurité. La plupart des solutions au problème de la sécurité de la configuration dépendent d'une configuration manuelle de chaque machine, ce qui ne réjouira pas l'administrateur système débordé. On peut parfois compter sur la protection apportée par les couches basses (un réseau Ethernet dont les commutateurs protègent l'accès aux différents ports, par exempe). Autrement, il existe plusieurs techniques d'authentification comme SEND (RFC 3971) mais qui, comme le note le RFC, dépend d'une configuration d'une clé sur chaque machine, ce qui n'est pas pratique, surtout lorsque la machine visite plusieurs réseaux.