RFC 6731 : Improved Recursive DNS Server Selection for Multi-Interfaced Nodes

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 19 décembre 2012

Date de publication du RFC : Décembre 2012

https://www.bortzmeyer.org/6731.html

Lorsqu'une machine connectée à l'Internet a plusieurs interfaces réseaux, et apprend (par exemple en DHCP) l'existence d'un résolveur DNS différent sur chaque interface, lequel utiliser ? Le principe de base de cette nouvelle norme est qu'il faudrait garder en mémoire chaque résolveur et utiliser à chaque fois celui qui correspond à l'interface réseau qu'on va utiliser.

Autrefois, le problème ne semblait pas crucial : après tout, un résolveur DNS en vaut un autre, puisque tous donnent les mêmes réponses. Mais ce n'est plus le cas aujourd'hui. Par exemple, sur certaines interfaces, le résolveur DNS peut donner accès à des noms de domaine privés, strictement locaux, que les autres résolveurs ne connaissent pas. On ne peut donc pas se contenter de prendre l'un des résolveurs au hasard, ni même « le meilleur » résolveur car chacun peut avoir des caractéristiques uniques (par exemple, l'un des résolveurs connait le domaine privé `foo.example` et un autre un autre domaine privé `bar.example`). D'où cette règle de garder en mémoire tous les résolveurs connus. (Le RFC utilise le terme de **RDNSS** - "*Recursive DNS Servers*" - pour les résolveurs.)

Ce n'est pas tout de mémoriser tous les résolveurs, encore faut-il savoir lequel utiliser. C'est le rôle de nouvelles options DHCP normalisées ici, qui permettent de distribuer les méta-informations dont le client a besoin pour savoir de quel résolveur se servir.

Si on veut mieux comprendre le problème de ces machines ayant plusieurs interfaces réseaux, il faut lire le premier RFC du groupe de travail MIF <http://tools.ietf.org/wg/mif>, le RFC 6418[1], qui définit le problème. Un exemple d'une telle machine est un "*smartphone*" qui a accès à la fois à un réseau WiFi privé et à la 3G. Mais il y a aussi le cas où il n'y a qu'une seule interface physique, mais plusieurs virtuelles, par exemple un accès à l'Internet par le FAI local et un VPN vers le réseau sécurisé de l'entreprise. La partie la plus évidente du problème des machines à plusieurs interfaces est celui des noms privés (section 2.1). Imaginons que la société Foobar ait le domaine `foobar.example` et qu'un sous-domaine `internal.foobar.example` contienne les informations privées, spécifiques à l'entreprise et non visibles dans le DNS public. Cela se fait en configurant spécialement les résolveurs utilisés par les employés de l'entreprise. Par exemple, avec Unbound, pour un domaine nommé `42` et servi par `91.191.147.246` et `91.191.147.243`, cela serait :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc6418.txt`

```
stub-zone:
   name: "42"
   stub-addr: 91.191.147.246
   stub-addr: 91.191.147.243
```

Si un de ces employés est en voyage, connecté depuis l'hôtel, ayant activé le VPN, et que sa machine utilise le résolveur de l'hôtel, il ne pourra pas utiliser les noms en `internal.foobar.example`. (Vous pouvez vous dire « il faut utiliser le résolveur DNS de l'entreprise pour tout ». Mais il sera en général plus lent et il pourra avoir ses propres limites, par exemple ne pas pouvoir résoudre `facebook.com` car le résolveur de l'entreprise censure ce nom.)

Notez aussi que le fait que l'application de ce RFC permettre de gérer les noms de domaines privés (non rattachés à l'arbre principal du DNS) ne signifie **pas** que ces noms privés soient une bonne idée, approuvée par l'IETF. Ce point avait été une des grosses controverses dans le groupe de travail. La section 6 donne à ce sujet des conseils pour minimiser les risques associés à ces noms privés. (En gros, les choisir uniques, par exemple en les suffixant du nom de l'organisation, pour que tout le monde ne nomme pas ses machines dans `.private` ou `.lan`.)

En général, le « bon » résolveur à utiliser dépend de l'interface de sortie et les techniques de ce RFC sont donc à utiliser en combinaison avec celles du RFC 4191 et du RFC 3442. Même si un nom de domaine est visible partout, il peut se résoudre en des adresses IP différentes selon l'interface réseau utilisé (cas célèbre de `smtp.orange.fr` qui ne donne pas le même résultat aux abonnés d'Orange et aux autres.) On peut encore citer le cas des extensions au DNS utilisées dans certains cas de coexistence IPv4-IPv6 (RFC 6147).

Il peut y avoir des cas encore plus complexes et notre RFC ne prétend pas tout résoudre (section 2.3).

Avant de décrire la solution standardisée, le RFC présente trois études de cas permettant d'illustrer un problème soluble (section 3). Des études de cas plus détaillées, avec examen des pratiques actuelles, figurent dans le RFC 6419.

D'abord, le cas d'un foyer connecté à l'Internet via un FAI ainsi qu'à un réseau privé, les deux connexions passant par le même CPE. Ce réseau privé utilise uniquement des noms dans un domaine qui n'apparait pas sur l'Internet (mettons `hidden.example`). Pour que tout marche, du point de vue de l'utilisateur, il faut envoyer les requêtes de résolution de noms au résolveur du réseau privé, si le nom se termine en `hidden.example` et au résolveur du FAI autrement. Le plus simple est qu'un relais DNS tourne sur le CPE et aiguille automatiquement les requêtes DNS vers le bon résolveur.

Autre scénario d'usage, celui mentionné plus tôt d'un "*smartphone*" connecte à la fois au réseau 3G et à un accès WiFi local. Pour des raisons financières, on souhaite utiliser le WiFi dans le plus grand nombre de cas. Mais on peut avoir besoin d'établir certaines connexions via la 3G, soit parce que certains services ne sont que là, soit pour des raisons de sécurité (à mon avis, le RFC surestime nettement la sécurité de la 3G mais c'est une autre histoire).

Le "*smartphone*" doit donc utiliser le DNS de l'accès WiFi par défaut mais avoir des règles (par exemple une liste de destinations sensibles qu'on ne veut contacter qu'en 3G) qui imposeront l'utilisation du résolveur DNS du fournisseur de 3G.

Enfin, dernière étude de cas, celui du "*road warrior*" doté d'un accès à un (ou plusieurs) VPN de son entreprise. Aussi bien pour des raisons de sécurité que d'accès à des noms de domaines privés, on veut

faire passer les accès au réseau interne de l'entreprise (requêtes DNS comprises) par le VPN. Par contre, pour consulter Facebook et YouTube, on va passer par l'accès par défaut.

La section 4 du RFC donne ensuite les règles exactes à suivre pour que tout se passe bien. Le principe essentiel est que la machine doit construire une liste de résolveurs (et pas uniquement garder le dernier annoncé) accessibles, en notant pour chacun les options envoyées par DHCP (nouvelles options décrites un peu plus loin), qui vont permettre de sélectionner le résolveur à utiliser pour une requête donnée. Par défaut, on considère que tous les résolveurs sont équivalents (donnent les mêmes résultats). Mais certains critères permettent de les distinguer :

— La confiance qu'on leur accorde (voir section 9.2 du RFC). C'est assez subjectif mais on peut penser, par exemple, qu'un résolveur atteint par un VPN IPsec vers son entreprise est plus digne de confiance que le résolveur d'un "*hotspot*" dans un café. Certains disent même que la 3G avec son fournisseur mobile est plus sûre (c'est l'exemple de pseudo-code dans l'annexe C, avec `IF_CELLULAR` préféré à `IF_WLAN`.) Cette confiance est à configurer dans le client, il n'y a pas de protocole pour la déterminer.
— La préférence indiquée en DHCP (voir les nouvelles options DHCP plus loin.).
— Les domaines privés qu'ils gèrent... ou pas.
— Le fait qu'ils valident avec DNSSEC ou pas. À noter que cela peut aussi dépendre des clés DNSSEC connues. Dans le cas le plus fréquent, le résolveur ne valide qu'avec la clé de la racine du DNS. Mais, s'il a d'autres clés configurées, il faut lire l'annexe B du RFC car c'est un peu plus compliqué.

Avec tous ces critères, on peut faire une sélection intelligente du résolveur. Ainsi, s'il faut résoudre un nom en `private.mycompany.example` et qu'un seul des résolveurs annonce ce domaine (les autres gérant l'arbre public du DNS), c'est clair qu'il faut utiliser ce serveur, les autres ne pourront pas répondre. Les règles de sélection exactes figurent dans la section 4.1. Si vous préférez lire le pseudo-code, l'annexe C contient une mise en œuvre de cet algorithme.

Les sections 4.2 et 4.3 décrivent les nouvelles options pour DHCPv4 (RFC 2131) et DHCPv6 (RFC 8415). Elles sont quasiment identiques en v4 et v6 donc je me limite à celle de v6 :

— Son code est 74 <`https://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xml#dhcpv6-parameters-2`> (146 en v4),
— Elle contient l'adresse IPv6 du résolveur et la **préférence** attribuée à ce serveur ("*High*", "*Medium*" ou "*Low*"),
— Elle contient une liste de domaines que le résolveur connait, par exemple `private.company.example`, (et `.` tout court s'il connait tous les domaines, ce qui est le cas d'un résolveur normal), ainsi qu'une liste de domaines correspondant à des préfixes IP pour les résolutions dans `ip6.arpa` (RFC 3596).

Un client DHCP qui connait notre RFC met une demande de cette option dans sa requête et reçoit en échange la liste ci-dessus. Attention, elle est limitée en taille par les contraintes de DHCP (section 4.4), il ne faut donc pas espérer y indiquer des centaines de domaines possibles.

Une fois qu'un résolveur a été sélectionné pour une requête DNS, il doit continuer à être utilisé si cette requête déclenche à son tour d'autres requêtes (section 4.7). Imaginons qu'on cherche l'adresse IPv6 de `portal.company.example`, on va faire une requête `AAAA portal.company.example` et si on obtient une réponse `portal.company.example CNAME www.example.com`, alors, la requête ultérieure `AAAA www.example.com` doit utiliser le même résolveur, pour être cohérente, au cas où `www.example.com` renvoie des adresses différentes selon le résolveur qui demande. Programmeurs, attention, cela implique de garder un état et de ne pas faire la sélection du résolveur de manière indépendante pour chaque requête.

De même qu'on garde une liste de résolveurs à utiliser, il faut aussi avoir autant de caches que d'interfaces, ou vider le cache entre deux requêtes, pour éviter qu'une information venant du cache n'« empoisonne » une résolution. Et il ne faut pas oublier, lorsqu'une interface réseau disparait, de vider

le cache des données apprises via cette interface (section 4.8). Le logiciel local de la machine devient donc assez compliqué !

Voilà, vous savez tout. Si le problème vous intéresse et que vous voulez creuser la question, l'annexe A de ce RFC présente « ce qui aurait pu être », les autres mécanismes qui auraient pu être choisis (comme « envoyer la requête sur toutes les interfaces réseau et prendre la première réponse qui revient »).

Actuellement, il semble que certains systèmes de Nokia et de Microsoft ("*Name Resolution Policy Table*") utilisent déjà ce mécanisme dans leurs téléphones.