RFC 8501 : Reverse DNS in IPv6 for Internet Service Providers

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 janvier 2019

Date de publication du RFC : Novembre 2018

https://www.bortzmeyer.org/8501.html

Le DNS permet d'avoir des enregistrements de type PTR, dans un sous-arbre spécifique sous `.arpa`, permettant d'associer un nom à une adresse IP. Les FAI et autres fournisseurs peuplent souvent ce sous-arbre afin que tous les clients aient un PTR, et donc que la « résolution inverse » (mal nommée, mais tant pis) rende un résultat. Avec IPv4, c'est facile. Si on gère un préfixe /22, on a 1 024 PTR à mettre, ce qui fait encore une assez petite zone DNS. Mais en IPv6 ? La zone ainsi pré-peuplée serait de taille colossale, bien au-delà des capacités des serveurs de noms. Il faut donc trouver d'autres solutions, et c'est ce que décrit ce RFC.

Deux exemples d'enregistrement PTR en IPv6, pour commencer, vus avec dig :

```
% dig +short -x 2a00:1450:4007:816::2005
par10s33-in-x05.1e100.net.
```

(1E100 = Google). Et un autre, avec tous les détails :

```
% dig -x 2001:67c:2218:e::51:41
...
;; QUESTION SECTION:
;1.4.0.0.1.5.0.0.0.0.0.0.0.0.0.0.e.0.0.0.8.1.2.2.c.7.6.0.1.0.0.2.ip6.arpa. IN PTR

;; ANSWER SECTION:
1.4.0.0.1.5.0.0.0.0.0.0.0.0.0.0.e.0.0.0.8.1.2.2.c.7.6.0.1.0.0.2.ip6.arpa. 600 IN PTR epp.nic.fr.
```

Et les mêmes, vus avec le "*DNS looking glass*" <https://www.bortzmeyer.org/dns-lg.html> : et .

Pourquoi mettre ces enregistrements PTR ? Le RFC 1912 [1], section 2.1, dit que ce serait bien de le faire (≪ "*For every IP address, there should be a matching PTR record in the in-addr.arpa domain*" ≫) mais ne donne pas de raison, à part que certains services risquent de vous refuser si vous n'avez pas ce PTR. En fait, il y a deux sortes de vérification que peut faire un service distant auquel vous vous connectez :

— Vérifier que votre adresse IP, celle du client, a un PTR, et refuser l'accès distant sinon. C'est relativement rare (et, à mon avis, assez bête).
— Vérifier que, **s'il y a un PTR**, il pointe vers un nom qui à son tour pointe vers l'adresse IP du client (et attention si vous programmez cela : on peut avoir plusieurs PTR, et les noms qu'ils indiquent peuvent avoir plusieurs adresses IP). Ce test est plus fréquent et bien plus justifié (vous n'êtes pas obligé d'avoir un PTR mais, si vous en avez un, il doit être cohérent). Notez que, par exemple, Postfix fait ce test (on obtient des messages du genre ≪ postfix/smtpd[818] : warning : hostname ppp-92-39-138-98.in-tel.ru does not resolve to address 92.39.138.98 : Name or service not known ≫, et de tels messages sont fréquents, indiquant que beaucoup d'administrateurs réseaux sont négligents) et que ce test n'est apparemment pas débrayable. Pour configurer les conséquences de ce test, voyez les paramètres `reject_unknown_client_hostname` et `reject_unknown_reverse_client_hostname`. Pour OpenSSH, c'est l'option `UseDNS` qui décide des conséquences du test.

La deuxième vérification peut se faire ainsi (en pseudo-code sur le serveur) :

```
if is_IPv4(client_IP_address) then
    arpa_domain = inverse(client_IP_address) + '.in-addr.arpa'
else
    arpa_domain = inverse(client_IP_address) + '.ip6.arpa'
end if
names = PTR_of(arpa_domain)    # Bien se rappeler qu'il peut y avoir plusieurs PTR
for name in names loop
    if is_IPv4(IP_address) then
        addresses = A_of(name) # Bien se rappeler qu'il peut y avoir plusieurs A ou AAAA
    else
        addresses = AAAA_of(name)
    end if
    if client_IP_address in addresses then
            return True
    end if
end loop
return False
```

Est-ce une bonne idée d'exiger un enregistrement PTR ? Le RFC 1912 le fait, mais il est vieux de 22 ans et est nettement dépassé sur ce point. Cette question des enregistrements PTR, lorsqu'elle est soulevée dans les forums d'opérateurs ou bien à l'IETF, entraine toujours des débats passionnés. Un projet de RFC avait été développé (`draft-ietf-dnsop-reverse-mapping-considerations`) mais n'avait jamais abouti. Mais ce qui est sûr est que ces PTR sont parfois exigés par des serveurs Internet, et qu'il faut donc réfléchir à la meilleure façon de les fournir.

En IPv4, comme indiqué dans le paragraphe introductif, c'est simple. On écrit un petit programme qui génère les PTR, on les met dans un fichier de zone, et on recharge le serveur DNS faisant autorité. Si le FAI `example.net` gère le préfixe IP `203.0.113.0/24`, la zone DNS correspondante est `113.0.203.in-addr.arpa`, et on met dans le fichier de zone quelque chose comme :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc1912.txt`

```
1   IN   PTR    1.north.city.example.net.
2   IN   PTR    2.north.city.example.net.
3   IN   PTR    3.north.city.example.net.
4   IN   PTR    4.north.city.example.net.
...
```

Et, si on est un FAI sérieux, on vérifie qu'on a bien les enregistrements d'adresse dans la zone `example.net` :

```
1.north.city IN A 203.0.113.1
2.north.city IN A 203.0.113.2
3.north.city IN A 203.0.113.3
4.north.city IN A 203.0.113.4
...
```

Mais en IPv6 ? Un PTR a cette forme (dans le fichier de zone de `a.b.b.a.2.4.0.0.8.b.d.0.1.0.0.2.ip6.arpa`) :

```
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR   1.north.city.example.net.
```

Et l'enregistrement d'adresse :

```
1.north.city IN AAAA 2001:db8:42:abba::1
```

Mais il faudrait autant de PTR que d'adresses possibles. Générer automatiquement la multitude de PTR qu'il faudrait, même pour un simple /64, est inenvisageable. Même si un programme arrivait à le faire (comme le note le RFC, en écrivant 1 000 entrées par seconde, il faudrait presque 600 millions d'années pour un pauvre /64), le serveur de noms ne pourrait pas charger une zone aussi grosse.

La section 2, le gros du RFC, décrit les solutions possibles. Elles n'ont pas les mêmes propriétés en terme de passage à l'échelle, conformité aux normes et disponibilité dans les logiciels existants, et, disons-le tout de suite, aucune n'est parfaite.

Première solution, la plus brutale, répondre NXDOMAIN (« ce nom n'existe pas »). C'est simple à faire et ça marche bien, quelle que soit la taille du préfixe. Il suffit d'un fichier de zone vide (juste les enregistrements SOA et NS) pour mettre en œuvre cette solution.

Après tout, c'est la réponse la plus honnête : pas d'information disponible. (Attention, on parle bien de répondre NXDOMAIN, et **surtout pas** de ne pas répondre. Ne pas répondre va laisser le client DNS attendre, peut-être très longtemps, alors qu'un NXDOMAIN va tout de suite lui dire qu'il n'y a pas de réponse.) Cela ne suit pas le RFC 1912, mais dont on a vu qu'il est très contestable sur ce point. Par contre, certains services seront refusés (à tort, je pense, mais la légende « s'il n'a pas de PTR, c'est un méchant » est très répandue) au client qui n'a pas de PTR.

Deuxième solution, profiter des jokers du DNS (RFC 4592) pour avoir une réponse identique pour toutes les adresses IP (cf. RFC 4472). C'est simple à faire et ça passe bien à l'échelle. Le fichier de zone va être :

```
*  IN PTR one-host.foo.bar.example.
```

Le principal inconvénient est que la requête en sens inverse (du nom - ici `one-host.foo.bar.example` - vers l'adresse IP) ne donnera pas un bon résultat, puisqu'il n'y a qu'un seul nom, ce qui plantera les programmes qui vérifient la cohérence des PTR et des A/AAAA.

Troisième solution, laisser les machines mettre à jour elle-mêmes le DNS, pour y mettre le PTR. La machine qui vient d'obtenir une adresse IP (que ce soit par SLAAC, DHCP ou un autre moyen), va mettre à jour enregistrements AAAA et PTR dans un serveur DNS dynamique. Cette fois, c'est compliqué à réaliser. Pour un gros FAI, il y aura plusieurs serveurs DNS, à des fins de redondance, il faudra que les machines clientes s'adressent au bon (il n'y a pas de mécanisme standard pour cela), s'il y a beaucoup de clients DNS, le serveur va souffrir, authentifier ces requêtes de mise à jour du DNS va être compliqué...Sans compter le risque qu'un utilisateur facétieux ne mette un PTR...« créatif » ou tout simplement un nom déjà utilisé.

Notez que, à la maison ou dans au autre petit réseau local, ce n'est pas forcément la machine terminale qui pourrait faire la mise à jour DNS. Cela pourrait se centraliser au niveau du routeur CPE (la "*box*").

Dans ce cas, on pourrait même imaginer que le FAI délègue deux noms de domaine à ce routeur CPE, un nom pour les requêtes d'adresses (mettons `customer-7359.city.example.net`) et un pour les requêtes inverses (mettons `6.a.2.1.6.a.a.b.8.b.d.0.1.0.0.2.ip6.arpa`). Dans ce cas, ce serait de la responsabilité du routeur CPE de répondre pour ces noms, d'une manière configurable par l'utilisateur. Si le FAI fournit la "*box*", comme c'est courant en France, il peut s'assurer qu'elle est capable de fournir ce service. Autrement, il faut ne faire cette délégation que si l'utilisateur a cliqué sur « oui, j'ai un serveur DNS bien configuré, déléguez-moi », ou bien en utilisant une heuristique (si un client demande une délégation de préfixe IP, on peut estimer qu'il s'agit d'un routeur et pas d'une machine terminale, et elle a peut-être un serveur DNS inclus). Notez que le RFC 6092 dit qu'un routeur CPE ne doit pas avoir, par défaut, de serveur DNS accessible de l'extérieur, mais cet avis était pour un résolveur, pas pour un serveur faisant autorité.

Enfin, la mise à jour dynamique du DNS peut aussi être faite par le serveur DHCP du FAI, par exemple en utilisant le nom demandé par le client (RFC 4704). Ou par le serveur RADIUS.

Quatrième solution, déjà partiellement évoquée plus haut, déléguer le DNS à l'utilisateur. Cela ne marche pas en général avec l'utilisateur individuel typique, qui n'a pas de serveur DNS, ni le temps ou l'envie d'en installer un. Une solution possible serait d'avoir une solution toute prête dans des offres de boitiers faisant tout comme le Turris Omnia.

Enfin, cinquième et dernière possibilité, générer des réponses à la volée, lors de la requête DNS. Un exemple d'algorithme serait, lors de la réception d'une requête DNS de type PTR, de fabriquer un nom (mettons `host-56651.customers.example.net`), de le renvoyer **et** de créer une entrée de type AAAA pour les requêtes portant sur ce nom. Ces deux entrées pourraient être gardées un certain temps, puis nettoyées. Notez que cela ne permet pas à l'utilisateur de choisir son nom. Et que peu de serveurs DNS savent faire cela à l'heure actuelle. (Merci à Alarig Le Lay pour m'avoir signalé le module qui va bien dans Knot <`https://www.swordarmor.fr/knot-reverse-automatique-et-dnssec.html`>.)

Cette technique a deux autres inconvénients. Si on utilise DNSSEC, elle impose de générer également les signatures dynamiquement (ou de ne pas signer cette zone, ce qui est raisonnable pour de l'information non-critique). Et l'algorithme utilisé doit être déterministe, pour donner le même résultat sur tous les serveurs de la zone.

La section 3 de notre RFC discute la possibilité d'un avitaillement du DNS par l'utilisateur final, via une interface dédiée, par exemple une page Web (évidemment authentifiée). Voici par exemple l'interface de Linode :

Notez qu'on ne peut définir un PTR qu'après avoir testé que la recherche d'adresses IP pour ce nom fonctionne, et renvoie une des adresses de la machine. Chez Gandi, on ne peut plus changer ces enregistrements depuis le passage à la version 5 de l'interface, hélas. Mais ça fonctionne encore tant que la version 4 reste en service :

L'interface de Free (qui ne gère qu'IPv4) est boguée : elle affiche le nom choisi mais une requête PTR donne quand même le nom automatique (du genre `lns-bzn-x-y-z.adsl.proxad.net`, d'autant plus drôle qu'il concerne un abonnement via la fibre).

La section 4 tente de synthétiser le problème et les solutions. Considérant qu'il y a six utilisations typiques des enregistrements PTR :

— Accepter ou rejeter du courrier, dans le cadre de la lutte anti-spam,
— Servir des publicités, en utilisant le PTR comme source de géolocalisation,
— Accepter ou rejeter des connexions SSH (en supposant, ce qui est très contestable, qu'un PTR avec validation de l'adresse est un bon indicateur d'un client SSH « sérieux »),
— La journalisation, en notant aussi bien le nom que l'adresse IP du client ; attention, noter le nom seul serait une erreur, puisqu'il est entièrement contrôlé du côté du client (l'administrateur de `2001:db8:42::1` peut, via son contrôle de `2.4.0.0.8.b.d.0.1.0.0.2.ip6.arpa`, mettre un PTR indiquant `www.google.com`),
— traceroute (à mon avis l'une des rares utilisations vraiment valables des enregistrements PTR),
— La découverte de services, suivant le RFC 6763.

Considérant ces usages, la meilleure solution serait la délégation du DNS à l'utilisateur, pour qu'il ait un contrôle complet, mais comme c'est actuellement irréaliste la plupart du temps, le NXDOMAIN est parfois la solution la plus propre. Évidemment, comme toujours, « cela dépend » et chaque opérateur doit faire sa propre analyse (rappelez-vous qu'il n'y a pas de consensus sur ces enregistrements PTR). Personnellement, je pense que tout opérateur devrait fournir à ses utilisateurs un moyen (formulaire Web ou API) pour mettre un enregistrement PTR.

Et pour terminer, la section 5 de notre RFC décrit les questions de sécurité et de vie privée liées à ces questions de résolution « inverse ». Parmi elles :

— Certaines personnes affirment bien fort que la présence d'un enregistrement PTR signifie quelque chose (par exemple que l'administrateur réseaux est sérieux/qualifié) mais c'est contestable.
— Ces enregistrements PTR, en l'absence de DNSSEC, n'ont qu'une valeur limitée pour la sécurité, puisqu'ils peuvent être manipulés, ajoutés ou détruits par un tiers ; tester ces enregistrements via un résolveur non-validant n'est pas très sérieux,
— Attention à la vie privée : trop d'information dans un PTR peut poser des problèmes (cf. RFC 8117),
— Lorsque l'utilisateur peut choisir la chaîne de caractères mise dans le PTR, son inventivité peut poser des problèmes (`le-ministre-machin-est-un-voleur.example.net`, ce qui peut valoir des ennuis juridiques au FAI).

Et si vous préférez lire en espagnol, Hugo Salgado l'a fait ici <`https://hugo.salga.do/post/180619110626/rfc8501-el-dns-reverso-en-ipv6-para-proveedores`>.