RFC 1996 : A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 27 septembre 2013

Date de publication du RFC : Août 1996

https://www.bortzmeyer.org/1996.html

Avant ce RFC, il n'existait pas de mécanisme dans le DNS pour prévenir les serveurs esclaves de la disponibilité de nouvelles données chez le serveur maître. Il fallait attendre que l'esclave contacte le maître (mécanisme de "*polling*"). Depuis notre RFC 1996 [1], un serveur maître peut prévenir ses esclaves avec un message NOTIFY, entraînant ainsi une mise à jour plus rapide des zones DNS.

Avant cela, le rythme de mise à jour était contrôlé par le champ Refresh de l'enregistrement SOA. Ce champ indiquait à quel rythme l'esclave devait contacter le maître à la recherche de nouvelles données. En moyenne, donc, le temps de mise à jour de tous les serveurs faisant autorité (maître et esclaves) était de la moitié du Refresh. Par exemple, la zone eu.org a un Refresh de 3 600 secondes :

```
% dig SOA eu.org
...
;; ANSWER SECTION:
eu.org. 86400 IN SOA ns.eu.org. hostmaster.eu.org. (
2013092601 ; serial
3600       ; refresh (1 hour)
1800       ; retry (30 minutes)
604800     ; expire (1 week)
86400      ; minimum (1 day)
)
...
```

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc1996.txt

Ce qui fait que les esclaves testeront le maître à intervalles d'une heure, lui demandant s'il a des nouvelles données depuis le numéro de série 2013092601. Si le maître répond aux requêtes SOA de ces esclaves avec un numéro de série plus récent, l'esclave transférera la zone (RFC 5936). Le problème est qu'on peut attendre longtemps. Dans le pire cas (si l'esclave a testé le maître juste avant que ce dernier ne soit mis à jour), on attendra une heure. La synchronisation entre serveurs faisant autorité (maîtres et esclaves) contribue donc au délai total de réjuvénation <https://www.bortzmeyer.org/dns-propagation.html>.

Le message NOTIY complète ce mécanisme de "*polling*" par un mécanisme d'interruption. Le maître envoit ce message à ses esclaves dès la mise à jour, et ceux-ci testent immédiatement.

À noter que le graphe des relations entre serveurs faisant autorité n'est pas forcément composé que d'un maître et d'esclaves transférant depuis le maître. On peut avoir des configurations plus complexes avec des esclaves transférant depuis un autre esclave, plusieurs maîtres, etc (c'est d'ailleurs pour cela que l'ancienne terminologie de serveur primaire et secondaires a été abandonnée).

La section 3 décrit le NOTIFY. Les messages DNS ont un champ nommé "*Opcode*" (section 4.1.1 du RFC 1035) dont les valeurs possibles sont dans un registre IANA <https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-5>. Le principal "*opcode*" rencontré dans la nature, et de loin, est le 0, QUERY, indiquant une requête DNS normale. NOTIFY est un autre "*opcode*" possible, de numéro 4. Lorsqu'un serveur a des données nouvelles, il envoie un message NOTIFY à tous ses esclaves, message auquel les esclaves répondront, pour rassurer le maître sur la bonne réception de ses informations. Autrement, le maître réessaiera (les NOTIFY, comme la plupart des messages DNS, sont transportés sur UDP et peuvent donc se perdre), un certain nombre de fois (le RFC recommande cinq fois). Le message du maître **peut** aussi contenir les nouvelles données. Dans les exemples ci-dessous, les maîtres envoient le nouveau SOA de la zone. Si le message avec un nouveau SOA est bien reçu par l'esclave, celui-ci se comporte comme si le délai Refresh était écoulé : il interroge le maître sur son numéro de série (les NOTIFY ne sont pas authentifiés et peuvent donc être trompeurs, cf. section 5) et, s'il y a bien eu mise à jour, transfère la zone.

La section 4 du RFC donne quelques exemples, mais j'ai plutôt inclus les miens. Tout d'abord, un serveur maître sur NSD. Sa configuration pour la zone bortzmeyer.42 comprendra la liste des esclaves à notifier (ici, un seul) :

```
zone:
      name: "bortzmeyer.42"
zonefile: "primary/bortzmeyer.42"
notify: 204.62.14.153 NOKEY
```

Maintenant, le serveur a de nouvelles données. Au moment où l'administrateur tape nsdc reload, le serveur envoie un NOTIFY que tcpdump montre ainsi :

```
22:58:53.934862 IP (tos 0x0, ttl 55, id 0, offset 0, flags [DF], proto UDP (17), length 59)
    217.70.190.232.51962 > 204.62.14.153.53: [udp sum ok] 32223 notify [b2&3=0x2400] SOA? bortzmeyer.42. (31
22:58:53.935055 IP (tos 0x0, ttl 64, id 26939, offset 0, flags [none], proto UDP (17), length 59)
    204.62.14.153.53 > 217.70.190.232.51962: [bad udp cksum 0x733f -> 0x1d3a!] 32223 notify*- q: SOA? bortzm
```

Le maître a notifié, l'esclave a répondu positivement.

Avec BIND, il n'est pas nécessaire de lister les serveurs esclaves, il les trouve par défaut dans l'enregistrement NS de la zone (contrairement à NSD, BIND a un résolveur interne). On peut compléter cette liste (ajouter des esclaves) avec la directive `also-notify`. Voici une notification envoyée par BIND :

```
23:11:40.781190 IP6 (hlim 55, next-header UDP (17) payload length: 100) 2001:67c:2218:3::1:4.1396 > 2605:4500:2:
23:11:40.781462 IP6 (hlim 64, next-header UDP (17) payload length: 37) 2605:4500:2:245b::42.53 > 2001:67c:2218:3
```

On trouve quelques enregistrements de trafic DNS avec `NOTIFY` sur pcapr <`https://www.bortzmeyer.org/pcapr.html`> : <`http://www.pcapr.net/browse?q=dns+notify`> (avec quelques faux positifs, aussi).

Si on veut envoyer à la main un `NOTIFY`, à des fins de test ou de déboguage, NSD a un outil utile, la commande `nsd-notify` :

```
% nsd-notify -z bortzmeyer.42 ns3.example.net
```

Si l'esclave n'est pas configuré pour recevoir des notifications de ce maître, NSD répond avec un refus :

```
[1379966010] nsd-notify[3346]: warning: bad reply from ns3.example.net \
    for zone bortzmeyer.42., error response REFUSED (5).
```

Alors que le RFC recommandait plutôt d'ignorer ce message `NOTIFY` inattendu. La configuration dans NSD pour accepter les notifications se fait avec la directive `allow-notify` :

```
allow-notify: 217.70.190.232 NOKEY
```

(Si vous voulez authentifier les `NOTIFY`, voyez mon autre article sur TSIG <`https://www.bortzmeyer.org/tsig-sans-bind.html`>.)