RFC 5001 : DNS Name Server Identifier Option (NSID)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 1 septembre 2007. Dernière mise à jour le 16 avril 2018

Date de publication du RFC : Août 2007

https://www.bortzmeyer.org/5001.html

Cette nouvelle option du DNS permet d'identifier le serveur de noms physique auquel on parle, dans le cas où plusieurs machines servent sous la même adresse IP. Elle a vocation à remplacer l'ancien `hostname.bind`.

En application du cahier des charges du RFC 4892 [1], ce très court RFC normalise une nouvelle utilisation d'EDNS pour transporter une chaîne de bits qui identifie une instance particulière d'un serveur de noms.

Aujourd'hui, en effet, comme l'explique le RFC 4892, il est fréquent qu'un même serveur de noms (par exemple `F.root-servers.net`, serveur de la racine) soit mis en œuvre par plusieurs machines physiques (des dizaines dans le cas de `F.root-servers.net`). En cas de défaillance d'une seule de ces machines, il est préférable de pouvoir détecter quelle machine était en cause (voir par exemple la panne de C-root <http://listes.cru.fr/sympa/arc/dns-fr/2007-06/msg00003.html>). Cela se faisait autrefois avec le nom spécial `hostname.bind` (un `id.server` a été proposé pour être moins lié à BIND). Cette méthode ayant plusieurs inconvénients (là encore, voir le RFC 4892), une nouvelle méthode était nécessaire.

NSID est simplement une option EDNS. Lorsqu'elle est présente dans la requête, le serveur mettra son identité dans la réponse.

En quoi consistera cette identité ? Ce fut le grand et vif débat au sein du groupe de travail dnsext <http://tools.ietf.org/wg/dnsext> de l'IETF. Certains operateurs de serveurs de noms ne souhaitaient pas en révéler trop, certains souhaitaient mettre de l'Unicode dans la réponse, etc. Le compromis a finalement été de décider que la chaîne de bits mise dans la réponse n'a aucune signification standard. Elle dépend entièrement de l'opérateur. Il pourra mettre un nom de machine (comme on le fait souvent aujourd'hui avec `hostname.bind`), un identificateur opaque, ou même une valeur différente à chaque fois (la section 3.1 détaille ces possibilités).

Voici une configuration de cette option sur un BIND 9.7 :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc4892.txt`

```
options {
    ...
    server-id "My super X-server";
};
```

et son résultat, lorsque le serveur est interrogé par dig :

```
% dig +nsid @ns1.example.org SOA example.org
...
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
; NSID: 4d 79 20 73 75 70 65 72 20 58 2d 73 65 72 76 65 72  \
           (M) (y) ( ) (s) (u) (p) (e) (r) ( ) (X) (-) (s) (e) (r) (v) (e) (r)
;; QUESTION SECTION:
...
```

Comme le contenu du NSID peut être absolument quelconque, dig affiche les valeurs des différents octets (et, pour être sympa, le caractère correspondant à ce code ASCII). Si on met une valeur en UTF-8 :

```
   server-id "Café-crème";
```

dig, n'affichera que les caractères ASCII :

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; NSID: 43 61 66 c3 a9 2d 63 72 c3 a8 6d 65  \
          (C) (a) (f) (.) (.) (-) (c) (r) (.) (.) (m) (e)
```

Avec nsd (depuis la version 3.2.5), la configuration est en hexadécimal (NSD 4 permet de mettre des chaînes ASCII). J'utilise ce petit programme (en ligne sur `https://www.bortzmeyer.org/files/string2hexa.py`) en Python pour faire la traduction. Mais on peut aussi, toujours en Python, faire simplement `python -c 'import sys;print sys.stdin.read().encode("hex")'` (merci à @alexpigne <http://twitter.com/alexpigne>) ou bien en classique shell Unix, utiliser `hexdump` avec `hexdump -v -e '/1 "%02X"'` (merci à @guguscat <http://twitter.com/guguscat>). Si vous le faites en une ligne de shell, n'oubliez pas le `-n` en argument à `echo`. En tout cas, voici un exemple :

```
% echo -n "ns1.example.net" | hexdump -v -e '/1 "%02X"'
6E73312E6578616D706C652E6E6574

# nsd.conf
nsid: 6E73312E6578616D706C652E6E6574
```

Si vous cherchez un exemple réel, les serveurs de l'AFNIC ont NSID :

```
% dig +nsid @c.nic.fr. SOA fr
...
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; NSID: 64 6e 73 2e 74 68 32 2e 6e 69 63 2e 66 72  (d) (n) (s) (.) (t) (h) (2) (.) (n) (i) (c) (.) (f) (r)
```

Ici, c'était une machine installée au Telehouse 2 (th2) à Paris. Mais c'est évidemment plus drôle avec une « machine » anycast, essayez depuis plusieurs endroits du réseau avec d.nic.fr. Une telle étude a été faite et publiée <https://labs.ripe.net/Members/stephane_bortzmeyer/using-atlas-udm-to-find-the-popular-instances-of-a-dns-anycast-name-server>. Autre exemple d'utilisation importante de NSID, sur le serveur L-root, décrit dans le RFC 7108.

Si vous voulez avoir une idée de ce qu'implique une mise en œuvre de ce RFC, vous pouvez regarder le « commit » 4cd33c... de GRONG <http://github.com/bortzmeyer/grong/commit/4cd33cf0758a5b9820090ca9443a4b40dead3fdd>. Voici son utilisation :

```
% grong-server -servername="ns1.local.bortzmeyer.org"
...
% dig +nsid @::1 nimportequoi
...
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; NSID: 6e 73 31 2e 6c 6f 63 61 6c 2e 62 6f 72 74 7a 6d 65 79 65 72 2e 6f 72 67  \
      (n) (s) (1) (.) (l) (o) (c) (a) (l) (.) (b) (o) (r) (t) (z) (m) (e) (y) (e) (r) (.) (o) (r) (g)
...
```

Le logiciel check-soa <https://framagit.org/bortzmeyer/check-soa> a une option -nsid qui peut servir simplement à voir les NSID des serveurs d'une zone :

```
 % check-soa -nsid tf
d.ext.nic.fr.
192.5.4.2: OK: 2222247941
2001:500:2e::2: OK: 2222247941
d.nic.fr.
194.0.9.1: OK: 2222247941 (NSID dns.ix1.nic.fr)
2001:678:c::1: OK: 2222247941 (NSID dns.ix1.nic.fr)
e.ext.nic.fr.
193.176.144.22: OK: 2222247941 (NSID ns-ext1a.sidn.nl)
2a00:d78:0:102:193:176:144:22: OK: 2222247941 (NSID ns-ext1a.sidn.nl)
f.ext.nic.fr.
194.146.106.46: OK: 2222247941 (NSID s2.par)
2001:67c:1010:11::53: OK: 2222247941 (NSID s2.sth)
g.ext.nic.fr.
194.0.36.1: OK: 2222247941 (NSID 1.lys.pch)
2001:678:4c::1: OK: 2222247941 (NSID 1.lys.pch)
```

Si vous voulez voir les requêtes NSID en Go, voici le commit <https://github.com/bortzmeyer/check-soa/commit/f960fbfabdc66c81171cc8b09562b7e426425de5>.

Le logiciel Blaeu <https://framagit.org/bortzmeyer/blaeu> de création et d'analyse de mesures des sondes RIPE Atlas <https://atlas.ripe.net/> peut également demander et afficher les NSID :

```
% blaeu-resolve --nsid --nameserver d.nic.fr yt
Nameserver d.nic.fr
[NSID: b'dns.lon.nic.fr'] : 1 occurrences
[NSID: b'dns.lyn.nic.fr'] : 1 occurrences
[NSID: b'dns.bru.nic.fr'] : 1 occurrences
[NSID: b'dns.fra.nic.fr'] : 1 occurrences
[NSID: b'dns.ix1.nic.fr'] : 1 occurrences
Test #12181106 done at 2018-04-16T13:48:38Z
```

L'analyse avec dnspython <http://www.dnspython.org/> est à cet endroit <https://framagit.org/bortzmeyer/blaeu/blob/master/blaeu-resolve#L281>.