RFC 9660 : The DNS Zone Version (ZONEVERSION) Option

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 13 octobre 2024

Date de publication du RFC : Octobre 2024

https://www.bortzmeyer.org/9660.html

Cette nouvelle option du DNS permet au client d'obtenir du serveur le numéro de version de la zone servie. (Et, non, le numéro de série dans l'enregistrement SOA ne suffit pas, lisez pour en savoir plus.)

Cela permettra de détecter les problèmes de mise à jour des serveurs faisant autorité <https://www.bortzmeyer.org/serveur-dns-faisant-autorite.html> si, par exemple, un des secondaires ne se met plus à jour. C'est surtout important pour l'"*anycast*", qui complique le déboguage. En combinaison avec le NSID du RFC 5001 [1], vous trouverez facilement le serveur qui a un problème. Cette nouvelle option ressemble d'ailleurs à NSID et s'utilise de la même façon.

Vous le savez, le DNS ne garantit pas que tous les serveurs faisant autorité <https://www.bortzmeyer.org/serveur-dns-faisant-autorite.html> d'une zone servent la même version de la zone au même moment. Si je regarde la zone .com à cet instant (14 juillet 2024, 09 :48 UTC) avec check-soa <https://framagit.org/bortzmeyer/check-soa>, je vois :

```
% check-soa com
a.gtld-servers.net.
2001:503:a83e::2:30: OK: 1720950475
192.5.6.30: OK: 1720950475
b.gtld-servers.net.
192.33.14.30: OK: 1720950475
2001:503:231d::2:30: OK: 1720950475
c.gtld-servers.net.
2001:503:83eb::30: OK: 1720950475
192.26.92.30: OK: 1720950475
```

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5001.txt

```
d.gtld-servers.net.
2001:500:856e::30: OK: 1720950475
192.31.80.30: OK: 1720950475
e.gtld-servers.net.
2001:502:1ca1::30: OK: 1720950475
192.12.94.30: OK: 1720950475
f.gtld-servers.net.
2001:503:d414::30: OK: 1720950475
192.35.51.30: OK: 1720950475
g.gtld-servers.net.
192.42.93.30: OK: 1720950475
2001:503:eea3::30: OK: 1720950475
h.gtld-servers.net.
192.54.112.30: OK: 1720950475
2001:502:8cc::30: OK: 1720950475
i.gtld-servers.net.
2001:503:39c1::30: OK: 1720950460
192.43.172.30: OK: 1720950475
j.gtld-servers.net.
2001:502:7094::30: OK: 1720950475
192.48.79.30: OK: 1720950475
k.gtld-servers.net.
192.52.178.30: OK: 1720950460
2001:503:d2d::30: OK: 1720950475
l.gtld-servers.net.
192.41.162.30: OK: 1720950475
2001:500:d937::30: OK: 1720950475
m.gtld-servers.net.
2001:501:b1f9::30: OK: 1720950475
192.55.83.30: OK: 1720950475
```

On observe que, bien que le numéro de série dans l'enregistrement SOA soit 1720950475, certains serveurs sont restés à 1720950460. Le DNS est « modérement cohérent » (RFC 3254, sur ce concept).

Dans l'exemple ci-dessus, check-soa a simplement fait une requête pour le type de données SOA (section 4.3.5 du RFC 1034). Une limite de cette méthode est que, si on observe des données d'autres types qui ne semblent pas à jour et que, pour vérifier, on fait une requête de type SOA, on n'a pas de garantie de tomber sur le même serveur, notamment en cas d'utilisation d'"*anycast*" (RFC 4786) ou bien s'il y a un répartiteur de charge avec plusieurs serveurs derrière. (D'ailleurs, dans l'exemple ci-dessus, vous avez remarqué que vous n'avez pas la même réponse en IPv4 et IPv6, probablement parce que vous arriviez sur deux instances différentes du nuage "*anycast*".) Il faut donc un mécanisme à l'intérieur même de la requête qu'on utilise, comme pour le NSID. C'est le cas du ZONEVERSION de ce RFC, qui permet d'exprimer la version sous forme d'un numéro de série (comme avec le SOA) ou par d'autres moyens dans le futur, puisque toutes les zones n'utilisent pas le mécanisme de synchronisation habituel du DNS. On peut par exemple avoir des zones entièrement dynamiques et tirées d'une base de données, ou bien d'un calcul.

Notez aussi que certaines zones, comme `.com`, changent très vite, et que donc, même si on tombe sur le même serveur, le numéro de série aura pu changer entre une requête ordinaire et celle pour le SOA. C'est une raison supplémentaire pour avoir le mécanisme ZONEVERSION.

Bref, le nouveau mécanisme (section 2 du RFC), utilise EDNS (section 6.1.2 du RFC 6891). La nouvelle option EDNS porte le numéro 19 <`https://www.iana.org/assignments/dns-parameters/dns-parameter`
`xml#dns-parameters-11`>. Encodée en TLV comme toutes les options EDNS, elle comprend une longueur et une chaine d'octets qui est la version de la zone. La longueur vaut zéro dans une requête (le client indique juste qu'il souhaite connaitre la version de la zone) et, dans la réponse, une valeur non nulle. La version est elle-même composée de trois champs :

— Le nombre de composants dans le nom de domaine pour la zone concernée. Si la requête DNS portait sur le nom de domaine `foo.bar.example.org` et que le serveur renvoie la version de la zone `example.org`, ce champ vaudra deux.
— Le type de la version. Différents serveurs faisant autorité utiliseront différentes méthodes pour se synchroniser et auront donc des types différents. Pour l'instant, un seul est normalisé, le 0 (alias SOA-SERIAL), qui est le numéro de série, celui qu'on trouverait dans le SOA, cf. section 3.3.13 du RFC 1035. La longueur sera donc de 6, les 4 octets de ce numéro, le type et le nombre de composants, un octet chacun. Un nouveau registre IANA <https://www.iana.org/assignments/dns-parameters/dns-parameters.xml#zoneversion-type-values> accueillera peut-être d'autres types dans le futur, suivant la procédure « spécification nécessaire » du RFC 8126. (Cette procédure nécessite un examen par un expert, et la section 7.2.1 guide l'expert pour ce travail.)
— La valeur à proprement parler.

Si vous voulez voir dans un pcap, regardez .

L'option ZONEVERSION n'est renvoyée au client qui si celui-ci l'avait demandé, ce qu'il fait en mettant une option ZONEVERSION vide dans sa requête (section 3 du RFC). Si le serveur fait autorité pour la zone concernée (ou une zone ancêtre), et qu'il gère cette nouvelle option, il répond avec une valeur. Même si le nom demandé n'existe pas (réponse NXDOMAIN), l'option est renvoyée dans la réponse.

Comme les autres options EDNS, elle n'est pas signée par DNSSEC (section 8). Il n'y a donc pas de moyen de vérifier son authenticité et elle ne doit donc être utilisée qu'à titre informatif, par exemple pour le déboguage. (En outre, elle peut être modifiée en route, sauf si on utilise un mécanisme comme DoT - RFC 7858.)

Question mises en œuvre de cette option, c'est surtout du code expérimental pour l'instant, voir cette liste <https://github.com/huguei/rrserial> (pas très à jour ?). Personnellement, j'ai ajouté ZONEVERSION à Drink <https://framagit.org/bortzmeyer/drink/-/commit/f4e64c5eaff895f513168435e5 et écrit un article sur l'implémentation d'options EDNS <https://www.bortzmeyer.org/edns-option.html> (mais notez que l'option a changé de nom et de format depuis). Notez que, contrairement à presque toutes les options EDNS, ZONEVERSION est par zone, pas par serveur, ce qui est une contrainte pour le ou la programmeureuse, qui ne peut pas choisir la valeur avant de connaitre le nom demandé. Du côté des autres logiciels, NSD a vu un patch <https://github.com/huguei/nsd/tree/rrserial> (mais apparemment abandonné). Voici ce que voit dig actuellement (en attendant une intégration officielle de l'option) :

```
% dig +ednsopt=19 @ns1-dyn.bortzmeyer.fr dyn.bortzmeyer.fr SOA
...
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64655
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
...
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 1440
; OPT=19: 03 00 78 a5 08 cc ("..x...")
...

;; ANSWER SECTION:
dyn.bortzmeyer.fr. 0 IN SOA ns1-dyn.bortzmeyer.fr. stephane.bortzmeyer.org. (
2024081612 ; serial
...
```

Vous noterez que "03" indique trois composants (`dyn.bortzmeyer.fr`), "00" le SOA-SERIAL, et "78 a5 08 cc" égal 2024081612. Vous pouvez aussi tester ZONEVERSION avec le serveur de test `200.1.122.30` (un NSD modifié), avec le domaine `example.com`.