RFC 5936 : DNS Zone Transfer Protocol (AXFR)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 juin 2010

Date de publication du RFC : Juin 2010

https://www.bortzmeyer.org/5936.html

Parmi les protocoles qui forment le socle de l'Internet, une des particularités du DNS est la grande distance entre la norme officielle, vieille désormais de treize ans et jamais révisée intégralement, et la réalité. Celle-ci a nécessité a plusieurs reprises la publication de normes de clarification, détaillant ce que le RFC 1034[1] avait laissé dans l'ombre. Notre RFC 5936 continue cette tradition pour le cas des **transferts de zone** où un serveur de noms réclame à un serveur **faisant autorité** une copie complète de la zone (c'est-à-dire des données DNS). Le principal point vague était le cas où un serveur faisait également autorité pour des zones parentes ou filles de la zone transférée : devait-il tenir compte de ce qu'il avait appris à cette occasion pour « corriger » la zone transférée ? La réponse est clairement désormais **non** : le serveur doit transférer ce qu'on lui a donné comme contenu de la zone, pas ce qu'il a pu apprendre par ailleurs.

Un peu d'histoire d'abord : le transfert de zones (souvent désigné par le mnémonique AXFR) était officiellement normalisé dans des parties du RFC 1034 et RFC 1035 (section 4.3.5 du premier, sections 2.2, 3.2.3 et 4.2 du second). Diverses améliorations avaient été créés comme le transfert **incrémental** (IXFR) du RFC 1995 ou comme la notification immédiate du RFC 1996. Le protocole avait également été corrigé par la section 5.5 du RFC 2181. Bien des points étant flous dans les RFC originels, les programmeurs ont dû petit à petit concevoir une version non-officielle, mais plus précise, du protocole, et c'est elle que notre RFC décrit.

Quel est le problème qu'essaie de résoudre le transfert de zones ? Pour des raisons de fiabilité, toute zone doit être servie par au moins deux serveurs faisant autorité (et bien plus en pratique). Évidemment, il faut que ces serveurs disposent des mêmes informations, soient synchronisés. Il existe plusieurs méthodes pour cela. Parmi les TLD, par exemple, certains utilisent rsync, certains utilisent les mécanismes de réplication de la base de données sous-jacente (comme Slony pour PostgreSQL). Mais AXFR (et IXFR)

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc1034.txt

sont les seuls mécanismes DNS du lot. Aussi, bien que cela ne soit pas, en toute rigueur, obligatoire, tous les logiciels serveurs de noms mettent en œuvre cette technique (section 1.2 du RFC).

Focalisons nous sur `AXFR` car `IXFR` et `NOTIFY` ne sont pas traités par ce nouveau RFC. La section 2 de notre RFC 5936 reprend donc le travail et décrit intégralement le protocole. Un client DNS envoie une requête `AXFR` en mettant le `QTYPE` ("*Query TYPE*") à la valeur 252 (section 2.1) et en dirigeant la requête vers un serveur faisant autorité (jamais vers un résolveur). Avec dig, cela se fait ainsi (ici pour la zone `.uk`) :

```
% dig @ns1.nic.uk AXFR uk.
```

Le client doit en outre s'assurer de la valeur de certains champs (section 2.1.1). Il **peut** aussi ajouter une section additionnelle à la question (section 2.1.5) spécifiant l'utilisation d'EDNS (RFC 2671) et/ou d'une technique d'authentification comme TSIG (RFC 8945).

La section 2.2 décrit la réponse à cette requête. C'est une série de messages DNS (lorsqu'il est transporté sur TCP, chaque message DNS est composé d'une longueur, puis de données). Chaque message peut comporter plusieurs enregistrements. Le premier message **doit** commencer avec l'enregistrement SOA de la zone transférée et le dernier message se terminer avec le même SOA. À part cette règle, l'ordre des enregistrements ou des messages n'a pas de signification (ce n'est donc pas forcément l'ordre du fichier de zone). Voici une partie de la réponse à la requête faite plus haut. Elle ne comporte qu'un seul message, incluant les 195 enregistrements de cette petite zone. La réponse commence bien par le SOA de la zone transférée :

```
uk. 172800 IN SOA ns1.nic.uk. hostmaster.nic.uk. 1277504551 7200 900 2419200 172800
uk. 3600 IN RRSIG NSEC3PARAM 8 1 3600 20100709182231 20100625172231 44346 uk. dM777Q8sZ9sNshexmw08/cfUIAqzx
...
_nicname._tcp.uk. 172800 IN SRV 0 0 43 whois.nic.uk.
ac.uk. 172800 IN NS ns.uu.net.
...
british-library.uk. 172800 IN NS dns1.bl.uk.
british-library.uk. 172800 IN NS dns2.bl.uk.
co.uk. 172800 IN NS ns1.nic.uk.
...
u1fmklfv3rdcnamdc64sekgcdp05bbiu.uk. 172800 IN NSEC3 1 1 0 - 2D51R16SS5MQU6PCG2Q0L05JA7E20FC7 NS SOA RRSIG D
uk. 172800 IN SOA ns1.nic.uk. hostmaster.nic.uk. 1277504551 7200 900 2419200 172800
;; Query time: 430 msec
;; SERVER: 2a01:40:1001:35::2#53(2a01:40:1001:35::2)
;; WHEN: Sat Jun 26 18:33:50 2010
;; XFR size: 195 records (messages 1, bytes 6125)
```

La réponse se termine bien par le SOA de la zone transférée. Bien que dig ne l'affiche pas lors d'un transfert de zones, certaines options doivent être mises dans l'en-tête, par exemple AA pour "*Authoritative Answer*" (section 2.2.1).

Dans l'exemple ci-dessus, la section additionnelle était vide. Mais, si on authentifie le transfert avec TSIG (RFC 8945), ce n'est plus le cas. La signature de la zone transférée apparait alors dans cette section (section 2.2.5 ; mais dig l'affiche à la fin des réponses, ce qui est trompeur. Si on veut voir ce qui passe sur le réseau, il vaut mieux utiliser Wireshark.)

Cela, c'était le protocole. Maintenant, les questions les plus sérieuses posées par le transfert de zones portaient plutôt sur le **contenu** de ce qui est transféré (section 3). Le principe de base posé par notre nouveau RFC est que le serveur doit transférer fidèlement ce que lui-même a reçu (via un fichier de zone - la méthode traditionnelle - ou via un SGBD). Notez que cette règle semble un enfonçage de porte ouverte mais qu'elle ne l'est pas : par exemple, si la zone est très dynamique, fabriquée à la demande, alors un transfert de zones est tout simplement impossible (section 3.1). Pour prendre un exemple pour rire, c'est le cas de `rp.secret-wg.org`.

Mais le principal cas où l'application de la règle est délicate est celui cité dans la section 3.2 : que faire si le serveur qui fait autorité pour `foo.example` fait également autorité pour `bar.foo.example` et qu'on lui demande de transférer `foo.example` ? Normalement, les enregistrements NS des deux zones doivent être identiques. Mais, dans le monde réel, ce n'est pas toujours le cas. Si `foo.example` contient :

```
bar.foo.example.    IN   NS   ns1.example.net.
                    IN   NS   ns2.example.net.
```

(enregistrements qui ne font pas autorité mais sont nécessaire spour trouver les serveurs de la zone fille) et que `bar.foo.example` contient :

```
bar.foo.example.    IN   NS   ns1.example.net.
                    IN   NS   ns3.example.net.
```

(qui, cette fois, font autorité) quel jeu d'enregistrements doit renvoyer le serveur lorsqu'on lui demande `foo.example` ? Ce n'est pas dit clairement dans les RFC 1034 (sa section 4.2.1 est ambigue) et RFC 1035 et les serveurs de noms ont souvent eu des comportements différents sur ce point. Or, celui-ci est crucial, par exemple pour des zones signées avec DNSSEC (par exemple, les enregistrements NSEC et NSEC3 ne seront pas les mêmes, puisque le nom suivant, dans chacune des deux zones, sera différent). Notre nouveau RFC tranche : on envoie ce que contenait la zone transférée, pas ce qu'on a pu apprendre parce qu'on sert d'autres zones. Dit autrement, `AXFR` doit être exactement équivalent à un rsync du fichier de zone.

Pourquoi l'ambiguité a t-elle été tranchée dans ce sens ? Il y a plusieurs raisons parmi lesquelles :

— Certains serveurs de `foo.example` peuvent faire autorité également pour `bar.foo.example` et d'autres pas. Si le serveur transférait les données « corrigées », les différents serveurs de `foo.example` enverraient donc des contenus différents lors d'un transfert.
— Le déboguage (ici, la détection de l'incohérence entre `foo.example` et `bar.foo.example`) serait bien plus difficile si on ne pouvait pas accéder aux données brutes stockées sur le serveur.

Même règle (transférer la zone qu'on a reçu, ne pas tenter de la « corriger » avec ce qu'on a appris dans d'autres zones) s'applique aussi à la **colle**, ces enregistrements d'adresses IP qui, sans faire autorité, doivent parfois être stockés dans la zone parente, pour permettre de trouver un serveur de noms qui figure dans une zone fille (section 3.3).

Il reste deux problèmes liés au contenu de la zone : la compression de données (RFC 1035, section 4.1.4) lors d'un transfert ne peut pas être la même que pour une réponse typique (section 3.4) car elle doit tenir compte de la casse (le transfert doit désormais transmettre les noms en respectant leur casse, ce qui n'est pas le cas lors d'une requête DNS classique). Et enfin, il y a le problème des noms masqués. Si un serveur permet la mise à jour dynamique (RFC 2136), une nouvelle délégation (la création dynamique d'enregistrements NS) peut masquer des anciens noms (section 3.5). Ainsi, si la zone contenait `www.toto.foo.example`, la création d'enregistrements NS pour `toto.foo.example` masque complètement `www.toto.foo.example`. Dans ce cas, le RFC est clair : le transfert de la zone doit inclure les noms masqués.

Une fois réglé le protocole et le contenu, reste la question du transport. La section 4 traite son cas : le transfert de zones a toujours nécessité l'utilisation de TCP (la section 4.2 explique pourquoi cela restera le cas). Mais le reste n'est pas clair : par exemple, peut-on réutiliser la connexion TCP pour un deuxième transfert de zones ? Les serveurs de noms esclaves font généralement une requête SOA avant de demander le transfert de zones, l'examen du numéro de série que contient le SOA leur permet de savoir si un transfert est nécessaire. Cette requête SOA peut-elle passer sur la même connexion TCP que le transfert, si celui-ci a lieu ? Il est difficile de « faire parler » les vieux RFC sur ce point. Le fait qu'ils n'exigent pas que le serveur copie la question ou bien le "*Query ID*" dans la réponse semble indiquer que l'idée était en effet qu'une connexion TCP ne serve qu'au transfert de zone, et à un seul. Toutefois, notre RFC 5936 choisit de trancher en autorisant explicitement l'utilisation d'une connexion TCP pour plusieurs requêtes (pas forcément toutes `AXFR`). C'est même l'option recommandée, pour économiser des ressources sur le serveur.

Lorsque les RFC originels sur le DNS avaient été écrits, on prêtait moins d'attention à la sécurité qu'aujourd'hui. Ces RFC ne mentionnaient donc pas la possibilité de restreindre, voire d'interdire le transfert de zones. Comme l'explique la section 5, cette possibilité est au contraire considérée comme vitale aujourd'hui. (Voir mon article « Récupérer une zone DNS <https://www.bortzmeyer.org/recuperer-zone-dns.html> ».) Le RFC demande que l'autorisation de transfert puisse se faire par adresses IP, et par des techniques de signature comme TSIG. La politique **par défaut** devrait être « pas de transfert ».

Voici un exemple avec le serveur nsd (version 3) qui, comme recommandé par le RFC, ne permet pas de transfert par défaut :

```
zone:
        name: "bortzmeyer.fr"
        zonefile: "primary/bortzmeyer.fr"
        provide-xfr: 192.134.7.248  NOKEY
```

Ici, la machine 192.134.7.248 (et elle seule) est autorisée à transférer la zone (directive `provide-xfr`). Une tentative depuis une autre adresse donnera, par exemple sur le serveur :

```
Jun 28 20:23:58 aetius nsd[25004]: axfr for zone bortzmeyer.fr. \
       from client 2a01:e35:8bd9:8bb0:38cf:6d60:b99:4d94 refused, no acl matches
```

Avec BIND, on s'y prend en général en définissant des ACL, ici une ACL nommée `my-nets` :

```
acl my-nets {
                203.0.113.128/26;
                127.0.0.0/8;
                2001:db8:3003::/48;
};
```

puis en les utilisant dans les paramètres de la zone :

```
zone "bortzmeyer.fr" {
    allow-transfer {
                my-nets;
        };
    ...
};
```

et une tentative par une autre machine sera rejetée, le serveur notant :

```
Jun 28 20:27:41 lilith named[20452]: \
       client 2a01:e35:8bd9:8bb0:38cf:6d60:b99:4d94#51619: \
       zone transfer 'bortzmeyer.fr/AXFR/IN' denied
```

Et l'intégrité des données transférées ? La section 6 se penche sur la question et spécifie qu'un transfert doit être complet, et avec succès, pour que la zone ainsi transférée soit utilisée (en d'autres termes, le transfert doit être atomique).

On l'a vu, l'ancienne norme était loin d'être claire sur tous les points. La nouvelle est, on l'espère, plus précise. Mais alors, n'y a t-il pas des risques de mauvaise entente entre anciens serveurs et nouveaux clients (ou le contraire) ? La section 7 du RFC est consacrée à ce problème de compatibilité. Elle commence par noter qu'il est difficile de donner des garanties, puisque la sous-spécification du transfert de zones fait que les anciens logiciels présentent une grande variété de comportements. Souvent, cet ancien comportement n'est pas facilement détectable automatiquement, donc la règle est qu'un nouveau logiciel qui souhaite interagir avec les anciens doit le faire via une option de configuration. Limitons nous au point de vue du serveur (section 7.1). Il ne peut pas savoir si le client accepte plusieurs enregistrements par message et l'interaction avec des clients qui n'y arrivent pas nécessite donc une option spécifique (mais qui ne doit pas être activée par défaut). (Pour le point de vue du client, voir la section 7.2.)