RFC 7871 : Client Subnet in DNS Queries

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 20 juin 2016. Dernière mise à jour le 13 juin 2022

Date de publication du RFC : Mai 2016

https://www.bortzmeyer.org/7871.html

Ce nouveau RFC décrit une option EDNS qui permet à un client DNS d'indiquer au serveur l'adresse IP d'origine de la requête DNS. Pourquoi diable ferait-on cela, au prix de la vie privée ? Cette option est surtout utile dans le cas de l'utilisation d'un gros résolveur DNS public (comme Google Public DNS <https://www.bortzmeyer.org/google-dns.html>) **et** d'un CDN.

Pour comprendre pourquoi, il faut se rappeler que le DNS ne fonctionne pas de bout en bout. La machine de M. Michu émet des requêtes DNS à un **résolveur <https://www.bortzmeyer.org/resolveur-dns.html>** (en général fourni par le FAI ou par le réseau local, mais certaines personnes, victimes du marketing, préfèrent les résolveurs publics comme OpenDNS, plus lointains et plus lents). Ce résolveur, à son tour, interroge les **serveurs faisant autorité <https://www.bortzmeyer.org/serveur-dns-faisant-autorite.html>**. Si M. Michu voulait se connecter à un site Web pour voir des vidéos de chats, il est possible qu'un nouvel acteur soit présent, le CDN (comme Akamai) qui héberge les dites vidéos. Les serveurs DNS faisant autorité pour le CDN renvoient souvent une adresse IP différente selon l'adresse IP du résolveur qui les interroge. Si l'adresse IP du client est au Sénégal, l'adresse IP du serveur de vidéos qui sera renvoyée sera celle du centre de données le plus « proche » du Sénégal. En temps normal, cela fonctionne plus ou moins. Bien sûr, le serveur DNS faisant autorité pour le CDN ne voit pas le « vrai » client, M. Michu, il voit le résolveur DNS utilisé mais les deux sont proches : si M. Michu est en Colombie, son résolveur DNS le sera aussi. Voici un exemple, vu avec les sondes RIPE Atlas, où on demande, au Sénégal (SN) puis en Colombie (CO) l'adresse IP de `www.elysee.fr`, hébergé sur un CDN états-unien :

```
% blaeu-resolve --country SN www.elysee.fr
[207.123.59.254 209.84.7.126 8.27.7.254] : 1 occurrences
[209.84.7.126 8.253.3.254 8.27.7.254] : 1 occurrences
[4.26.233.254 4.26.236.126 8.254.119.126] : 1 occurrences
[207.123.59.254 209.84.7.126 8.253.3.254] : 1 occurrences
[207.123.59.254 8.26.223.254 8.27.7.254] : 1 occurrences
Test #4106632 done at 2016-06-15T22:52:44Z

% blaeu-resolve --country CO www.elysee.fr
[192.221.116.253] : 3 occurrences
[205.128.71.253 4.27.25.125 8.27.155.126] : 1 occurrences
[206.33.52.253 209.84.20.126 8.253.16.126] : 1 occurrences
Test #4106633 done at 2016-06-15T22:52:46Z
```

On voit qu'on a obtenu des adresses IP très différentes et, espérons-le, adaptées à chaque pays.

Autre exemple, avec un service de PowerDNS, qui renvoyait (il semble ne plus marcher) une géolocalisation du client. Ici, j'utilise dig depuis deux machines différentes, une en France et l'autre aux États-Unis :

```
%  dig +short -t txt www.geo.powerdns.com
"bonjour france 2a01:db8:8bd9:85cb:21e:8cff:fe76:29b6/26"

% dig +short -t txt www.geo.powerdns.com
"hello USA 204.62.14.153/22"
```

On voit que le résolveur que j'utilise (qui, à chaque fois, était sur le réseau local de la machine cliente), a bien été géolocalisé. Si j'utilise un résolveur public :

```
%  dig @8.8.8.8 +short -t txt www.geo.powerdns.com
"bonjour france XX.YY.152.0/11"
```

Ici, cela a marché car Google Public DNS a des serveurs en France. Si cela n'avait pas été le cas, j'aurais été géolocalisé... quelque part ailleurs.

Tout change (section 1 du RFC) si on utilise un résolveur DNS public comme Verisign Public DNS <https://www.verisign.com/publicdns> ou bien Yandex DNS <https://dns.yandex.com/>. Dans ce cas, l'adresse IP du résolveur, vue par le serveur faisant autorité pour le CDN, n'a plus grand'chose à voir avec celle du vrai client, elle peut être très lointaine. Les serveurs DNS faisant autorité risquent donc de renvoyer une adresse IP de serveur Web qui ne soit pas optimale. Certes, tout marchera quand même mais ça sera plus lent alors qu'officiellement, le but des CDN est d'accélerer l'arrivée de la vidéo du chat (non, de François Hollande).

On voit que ce RFC résout donc un problème que n'a pas tout le monde. Seuls ceux qui se servent d'un résolveur public lointain, et qui visitent des sites Web hébergés sur un CDN (en général les gros sites commerciaux) auront le problème. C'est une des raisons qui expliquent que ce RFC a pas mal trainé (voyez cet article, qui date de plusieurs années <https://vincent.bernat.im/fr/blog/2014-bind-edns0-client-subnet.html>) à l'IETF : son intérêt n'est pas évident, et il y avait beaucoup de contestation. Mais l'IETF a finalement choisi de documenter cette option (qui est effectivement déployée), sans forcément la recommander.

Donc, comment est-ce que cela résout le problème décrit plus haut ? L'idée est de standardiser une option EDNS que les résolveurs publics ajouteront dans leur requête aux serveurs faisant autorité, et qui indiquera la « vraie » adresse IP d'origine (section 5 du RFC). Imaginons que M. Michu ait pour adresse IP `192.0.2.56` et qu'il utilise Google Public DNS. Ce dernier va transmettre la requête au CDN et ajoutera l'option "*Client Subnet*" (ce n'est donc pas M. Michu qui le fera). Le serveur DNS du CDN verra une requête arriver de, mettons <https://developers.google.com/speed/public-dns/faq#locations>, `74.125.17.1`. L'option EDNS "*Client Subnet*" contiendra le préfixe de l'adresse IP de M. Michu, `192.0.2.0/25`. Il saura alors qu'il doit adapter ses réponses, non pas au préfixe `74.125.17.0/24` (son client direct) mais au préfixe `192.0.2.0/25` (le vrai client). C'est donc une option entre résolveur et serveur faisant autorité, pas entre machine terminale et résolveur.

Le format de l'option est décrite dans la section 6 du RFC. Comme toutes les options EDNS (RFC 6891 [1]), elle est encodée en TLV : le type est 8 <https://www.iana.org/assignments/dns-parameters/dns-parameters.xml#dns-parameters-11>, la valeur est composée de :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc6891.txt`

— La famille d'adresses utilisée (IPv4 ou IPv6, l'exemple ci-dessus utilisait IPv4),
— La longueur du préfixe indiqué dans la requête DNS (25 bits dans l'exemple ci-dessus),
— La longueur significative du préfixe dans la **réponse** DNS (elle est mise à zéro dans les requêtes),
— Le préfixe lui-même (`192.0.2.0` dans l'exemple plus haut).

La section 7 du RFC décrit les détails du protocole. Le résolveur va mettre dans son cache les réponses du serveur faisant autorité. Normalement, l'information dans le cache est indexée par le nom de domaine (et le type de données demandé, mais je simplifie). Avec ECS ("*EDNS Client Subnet*"), l'information est indexée par le couple {nom de domaine, préfixe IP}. En effet, deux requêtes pour le même nom peuvent avoir donné des résultats différents selon le préfixe IP (c'est bien le but !).

Le résolveur qui met cette option doit choisir la longueur du préfixe envoyé. Il tient compte de deux choses :

— La capacité de son cache : si le préfixe est trop spécifique, le cache devra stocker davantage de données (imaginons en IPv6 une réponse différente par adresse IP : le cache pourrait avoir à stocker 2{}128 réponses par nom de domaine !)
— La protection de la vie privée des utilisateurs. ECS la menace déjà, n'aggravons pas les choses. En utilisant un préfixe assez général, on limite l'indiscrétion.

Mais l'affaire est un peu plus compliquée : le client d'origine (la machine de M. Michu, le "*stub resolver*") a pu mettre une option ECS elle-même (ce n'est pas courant aujourd'hui mais ça pourrait arriver). Dans ce cas, le résolveur doit en tenir compte et mettre comme longueur la plus courte entre celle demandée par le client (a priori pour protéger sa vie privée) et celle que le résolveur aurait choisi tout seul.

Si la requête reçue par le résolveur avait l'option ECS et une longueur de zéro, cela indique le souhait du client qu'on ne transmette **pas** son adresse IP du tout. Le résolveur doit évidemment respecter cette demande.

Et le serveur faisant autorité, que doit-il mettre dans sa réponse ? (S'il envoie des réponses différentes selon la source, **et** s'il gère l'option ECS, "*EDNS Client Subnet*" ; autrement, c'est simple, il ne fait rien de particulier.) Il met une option ECS dans la réponses, avec :

— La famille, la longueur du préfixe demandé, et le préfixe identiques à celui de la requête,
— Une longueur effective ("*scope prefix length*") qui indique pour quel préfixe la réponse est valable. Cette longueur effective peut être supérieure à la longueur demandée (le préfixe était trop général, pense le serveur), ou inférieure (le préfixe était inutilement spécifique, le serveur ne varie pas ses réponses à ce point).

(Notez que le RFC est bien plus détaillé que ce résumé, car il y a plein de cas rigolos. Je me suis limité à une présentation générale, je n'essaie pas de traduire tout le RFC.)

En recevant cette réponse, le résolveur va la mettre dans son cache, en tenant compte de la longueur du préfixe (et, bien sûr, le résolveur transmet la réponse à son client). Une réponse valable pour `2001:db8:43:bef::/64` ne pourra pas être utilisée pour le client `2001:db8:43:bed::1`. Quelle longueur sera utilisée pour indexer le cache ? Celle demandée ou bien celle effective ? Les régles exactes sont un peu complexes (il faut tenir compte de la longueur effective, mais aussi des limites du cache, qui ne veut pas stocker une réponse pour des préfixes trop spécifiques), je vous renvoie à la section 7.3.1 du RFC.

Les questions ultérieures des clients du résolveur pourront recevoir une réponse tirée du cache, sans repasser par les serveurs faisant autorité. Mais ECS impose d'organiser le cache différemment. Avec le DNS classique, si on a dans le cache la réponse à une question pour `cat.example.com` (je simplifie en ne tenant pas compte du type des données DNS), et qu'on reçoit une question pour ce nom, on peut répondre avec les données du cache. Avec ECS, le cache doit en plus tenir compte du préfixe stocké avec la réponse, et de l'adresse IP du client.

Et avec DNSSEC, ça se passe comment (section 9 du RFC) ? Les CDN ne signent pas leur zone en général. Mais s'ils s'y mettent (il le faudrait), il y aura quelques précautions à prendre.

Et avec le NAT ? Il n'y a normalement pas de problèmes, sauf évidemment si le résolveur est NATé et ne le sait pas : il mettrait, dans ce cas, une adresse privée dans l'option ECS, ce qui serait idiot.

Reste à regarder les problèmes de sécurité et notamment de vie privée. ECS diminue forcément votre vie privée. Il ajoute en effet une information qui n'était pas là sans lui (et le RFC 8165 dit bien que c'est mal). C'est pour limiter les dégâts que le RFC recommande aux résolveurs qui ajoutent une option "*Client Subnet*" de la limiter aux 24 premiers bits en IPv4 et aux 56 premiers en IPv6. Un résolveur qui connait bien la topologie de son réseau peut faire encore mieux : s'il sait que tous ses clients sont proches, et couverts par le même /20, il peut ainsi ne transmettre que les vingt premiers bits, sans diminuer l'intérêt du service.

Notez que, avec les clients DNS d'aujourd'hui, votre résolveur mettra votre adresse IP dans ses requêtes sortantes. On peut en sortir ("*opt-out*" en mettant une option ECS avec une longueur nulle) mais cela nécessite une action explicite (que ne permettent pas forcément les clients DNS actuels, notamment les "*stub resolvers*", ceux qui sont intégrés dans le système d'exploitation). Le RFC recommande donc que cette option ECS soit **dés**activée par défaut, en raison de ces risques.

L'option ECS peut être vue par les serveurs faisant autorité, mais également par les tiers qui espionnent le trafic. Pour empêcher cela, il faudra déployer des solutions de chiffrement du trafic DNS, comme celles sur lesquelles travaille le groupe DPRIVE <https://tools.ietf.org/wg/dprive>.

Un bon article sur les problèmes de vie privée liés à ECS est le « "*Understanding the Privacy Implications of ECS*" <https://astrolavos.gatech.edu/2016/07/07/ecs/> » de Panagiotis Kintis, Yacin Nadji, David Dagon, Michael Farrell et Manos Antonakakis. Un autre bon article sur cette question est celui de Frank Denis <https://00f.net/2013/08/07/edns-client-subnet/>.

Voyons maintenant cette option ECS en action. Elle est par exemple utilisée par Google dont le résolveur public ajoute cette option avant de l'envoyer aux serveurs faisant autorité. Mais on trouve une liste d'utilisateurs plus détaillée sur le site de promotion de la technologie <http://www.afasterinternet.com/participants.htm>.

Parmi les logiciels libres qui la mettent en œuvre, on note la bibliothèque getdns <https://getdnsapi.net/>. Ou bien le programme dig livré avec BIND. Voici un exemple avec la nouvelle option `+subnet` de dig (prise sur un BIND 9.11, l'option était déjà en 9.10) :

```
% dig +subnet=8.189.152.0/25 @ns-1568.awsdns-04.co.uk  A www.amazon.com
...
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; CLIENT-SUBNET: 8.189.152.0/25/0
...
;; ANSWER SECTION:
www.amazon.com.         60 IN A 54.239.25.192
...
```

Et ce que voit le serveur faisant autorité <https://www.bortzmeyer.org/serveur-dns-faisant-autorit
html> ? On peut le savoir en faisant tourner tcpdump sur un serveur faisant autorité qu'on contrôle, mais il y a plus simple, utiliser un domaine dont les serveurs faisant autorité renverront l'information ECS qu'ils ont reçu. J'en connais trois, qui répondent aux requêtes DNS de type `TXT` :

— _country.pool.ntp.org,
— whoami[Caractère Unicode non montré [2]].[Caractère Unicode non montré]fastly[Cara
Unicode non montré].[Caractère Unicode non montré]net et whoami[Caractère
Unicode non montré]6.[Caractère Unicode non montré]fastly[Caractère
Unicode non montré].[Caractère Unicode non montré]net,
— et mon ecs.dyn.bortzmeyer.fr (qui utilise le logiciel Drink <https://www.bortzmeyer.org/drink.html>).

Voici un exemple :

```
% dig ecs.dyn.bortzmeyer.fr TXT
...
;; ANSWER SECTION:
ecs.dyn.bortzmeyer.fr. 0 IN TXT "78.196.62.0/24"

;; Query time: 15 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: lun. juin 13 14:08:23 CEST 2022
```

Un service analogue, edns-client-sub.net renvoie sous forme d'un objet JSON les options ECS reçues, ainsi que la géolocalisation. Ici, mon résolveur n'envoie pas ECS :

```
% dig +short -t txt edns-client-sub.net
"{'ecs':'False','ts':'1447875683.94','recursive':{'cc':'FR','srcip':'XX.YY.152.187','sport':'37512'}}"
```

Mais Google, lui, le fait (et heureusement, sinon j'aurais été géolocalisé en Belgique, où se trouve le serveur de Google utilisé) :

```
% dig @8.8.8.8 +short -t txt edns-client-sub.net
"{'ecs_payload':{'family':'1','optcode':'0x08','cc':'FR','ip':'XX.YY.152.0','mask':'24','scope':'0'},
     'ecs':'True','ts':'1447875689.25','recursive':{'cc':'BE','srcip':'74.125.47.152','sport':'41735'}}"
```

Mais ce service ne semble plus fonctionner, en juin 2022.

Enfin, voici un exemple de code Python qui coupe ECS chez le résolveur, pour protéger la vie privée de l'utilisateur. Il utilise la bibliothèque dnspython <http://www.dnspython.org/> :

```
opt = dns.edns.ECSOption(address='', srclen=0) # Disable ECS (RFC 7871, section 7.1.2)
options = [opt]
message = dns.message.make_query(qname, dns.rdatatype.from_text(qtype),
       use_edns=True, options=options)
```

2. Car trop difficile à faire afficher par LaTeX