Le projet Net4D d'utilisation des classes DNS

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 6 octobre 2009

https://www.bortzmeyer.org/net4d.html

Le projet Net4D <http://net4d.org/index1.html> vise à utiliser un champ peu connu du DNS, la **classe**, pour permettre de créer autant d'« espaces de noms » distincts que de classes, afin de démocratiser la gestion des noms de domaine, actuellement verrouillée par le gouvernement états-unien. Ce n'est pas forcément une bonne idée que de « casser » les projets rigolos et originaux mais, comme les promoteurs de Net4D ont manifestement l'oreille des médias <http://www.lemonde.fr/archives/article/2009/10/02/les-65-000-concurrents-de-l-icann_1248113_0.html>, je vais y aller quand même.

D'abord, désolé, mais je ne peux pas m'empêcher d'un petit retour technique. Le DNS distribue des **enregistrements**, dont la **clé** est un nom de domaine. Chaque enregistrement a un **type** (par exemple `AAAA` pour les adresses IP, `MX` pour les relais de courrier, etc). Il a aussi une **valeur**, qui dépend du type (par exemple, la valeur d'un enregistrement de type `LOC` est une latitude et une longitude). Il a aussi une durée de vie et, surtout, ce qui nous intéresse ici, une **classe**. Le champ `class` était quasiment oublié car il a toujours la même valeur, `IN` (pour Internet). Mais il est toujours là et figure également dans la **question** que pose un client DNS (champ `QCLASS` pour "*Query Class*"). Tout ceci est précisé dans le RFC 1034 [1], notamment sections 3.6 et 3.7.1.

À la fin de cet article, je donne quelques exemples techniques. Mais, avant cela, revenons à la proposition de Net4D <http://net4d.org/index1.html>. Aujourd'hui, seules trois classes sont enregistrées à l'IANA <https://www.iana.org/assignments/dns-parameters> :

```
Registry:
Decimal
Hexadecimal    Name                            Reference
-------------  ------------------------------  ---------
0              Reserved                        [RFC5395]
1              Internet (IN)                   [RFC1035]
```

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

```
2               Unassigned
3               Chaos (CH)                      [Moon1981]
4               Hesiod (HS)                     [Dyer1987]
5-253           Unassigned
254             QCLASS NONE                     [RFC2136]
255             QCLASS * (ANY)                  [RFC1035]
256-65279       Unassigned
65280-65534     Reserved for Private Use        [RFC5395]
65535           Reserved                        [RFC5395]
```

J'ai déjà parlé de la classe `IN` pour Internet, numéro 1. Chaos (classe numéro 3) était un protocole réseau qui n'a guère eu de succès mais qui utilisait le DNS. Et Hesiod (numéro 4) était un mécanisme de distribution d'information sur les utilisateurs via le DNS (je l'avais déployé au CNAM mais, aujourd'hui, tout le monde utilise LDAP ou le vieux NIS).

Aujourd'hui, un nom comme `www.carlabrunisarkozy.org` est donc quasiment toujours de la classe `IN`. C'est tellement le cas qu'on ne pense plus en général à l'indiquer. Il n'existe pas de mécanisme standard dans les URL (comme `http://www.carlabrunisarkozy.org/`) pour indiquer une classe, des fonctions pour les programmeurs comme `getaddrinfo()` n'ont pas de paramètre pour la classe, etc.

L'idée de base de Net4D est d'allouer un certain nombre de classes, actuellement libres, et d'y créer des racines du DNS, avec des noms qui auraient une signification différente. Par exemple, supposons qu'on alloue la classe numéro 2 sous le nom `UN`, on pourrait, dans la configuration des serveurs de noms, créer un `www.carlabrunisarkozy.org` qui n'aurait rien à voir avec celui cité précédemment et qui pourrait avoir un enregistrement de type adresse qui pointe vers un tout autre service. (La procédure d'allocation de nouvelles classes, relativement souple, figure dans la section 3.2 du RFC 6195.)

La motivation est politique : il s'agit, comme l'indique le titre du publi-reportage du Monde `<http://www.lemonde.fr/archives/article/2009/10/02/les-65-000-concurrents-de-l-icann_1248113_0.html>`, d'avoir plein de nouvelles ICANN, pour que d'autres puissent jouer aux jeux politiciens qui occupent les fréquentes réunions de cette organisation. Un autre `.com` pourrait donc voir le jour, où `sex.com` pourrait être vendu à un autre, un autre `.fr` serait alloué à une autre organisation que l'AFNIC, etc. C'est pour cela que l'UIT, qui souffre d'être tenue à l'écart de la gouvernance d'Internet par le gouvernement états-unien, finance Net4D.

Maintenant, quels sont les problèmes de cette approche ? Il y en a trois, un technique, un politique et un de méthode.

Le problème technique est le suivant : bien sûr, les classes fonctionnent, le logiciel est déjà là et tout est déjà normalisé. Mais, sur l'Internet, il y a ce qui marche en théorie et ce qui marche en pratique : l'Internet est hélas très ossifié aujourd'hui. Des tas de programmes mal écrits avec les pieds sont fermement installés et ne laissent pas passer des choses parfaitement légales. D'innombrables boitiers intermédiaires sont sur le chemin des paquets et filtrent ce qu'ils ne comprennent pas. Comme l'ont fort bien expliqué Avri Doria et Karl Auerbach lors des discussions sur la liste `governance@lists.cpsr.org`, le manque flagrant de robustesse de beaucoup de composants de l'Internet, leur incapacité à gérer des situations légales mais inattendues, ne laisse pas beaucoup d'espoir (voir le message d'Auerbach `<http://lists.cpsr.org/lists/arc/governance/2009-09/msg00326.html>` et celui de Doria `<http://lists.cpsr.org/lists/arc/governance/2009-09/msg00331.html>`; notez qu'aucun des deux n'est un défenseur de l'ICANN, bien au contraire). Entre deux logiciels sérieux situés sur deux machines du même réseau, les classes vont marcher. Sur l'Internet en général, c'est bien plus difficile (il suffit de voir le temps qu'il a fallu pour que des **types** légaux comme `SRV`, passent à peu près partout).

Ce problème technique est d'autant plus sérieux que les promoteurs de Net4D traitent avec une désinvolture inquiétante les points sur lesquels il **faudra** déployer du code nouveau (comme un remplaçant de `getaddrinfo()`).

Bref, déployer les classes serait sans doute quasiment autant de travail que de déployer un système de nommage et de résolution complètement nouveau (DNS 2.0 ?).

Le problème politique est qu'utiliser un truc technique astucieux (comme les classes) pour résoudre un problème politique (la mainmise du gouvernement états-unien et des titulaires de propriété intellectuelle sur l'ICANN) est en général une mauvaise idée. Le problème de fond est politique, et aucune astuce technique ne permettra de le contourner. Évidemment, il est plus facile de rêver à se créer son monde idéal, plutôt qu'à s'attacher à changer celui qui existe...

Enfin, il y a un problème de méthode. L'UIT, on l'a vu, finance, avec l'argent public, la technique des classes. Ce n'est pas très compliqué que de configurer un serveur comme NSD pour charger une racine composée de plusieurs classes (par exemple la numéro 65280, prévue pour les essais), tester la délégation, les logiciels clients DNS courants, une version modifiée de `getaddrinfo()`, un site Web accessible via une classe autre que `IN`, etc. Ce serait un bon exercice dans un cours sur les réseaux à l'Université. Une entreprise sérieuse pourrait réaliser une telle étude en un temps, et pour une somme très raisonnable. Mais rien ne semble avoir été fait en ce sens (en tout cas rien n'a été publié). Tout le temps et l'argent ont été dépensés en relations publiques, dont l'article du Monde, qui ne donne la parole qu'à un seul point de vue, est un bon exemple. Cela n'augure pas bien du sérieux de la démarche.

Terminons avec un peu de technique. Une des rares utilisations aujourd'hui des classes autres que `IN` est pour détecter la version d'un serveur de noms (il existe désormais une autre méthode, dans le RFC 5001, mais elle est encore peu utilisée). Cela se fait en interrogeant le serveur pour le type `TXT`, la classe `CH`, et le nom `version.bind` :

```
% dig +short @f.root-servers.net CH TXT version.bind.
"9.5.1-P3"
```

Le domaine de tête `.bind` n'existe pas dans la classe `IN`.

Enfin, pour ceux qui s'intéressent aux bits sur le câble, pour décoder les paquets DNS <https://www.bortzmeyer.org/pcap-decodage-dns.html>, la classe figure dans la question, juste après le nom et le type. En C, le décodage ressemble donc à :

```
struct dns_packet {
...
    uint16_t    qtype, qclass;
...
/* sectionptr indicates the next byte in the packet */
decoded->qtype = ntohs(*((uint16_t *) sectionptr));
sectionptr += 2; /* Two bytes for the type */
CHECK_SECTIONPTR(2); /* Two bytes for the class */
decoded->qclass = ntohs(*((uint16_t *) sectionptr));
```