RFC 7393 : Using Port Control Protocol (PCP) to update dynamic DNS

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 8 novembre 2014

Date de publication du RFC : Novembre 2014

https://www.bortzmeyer.org/7393.html

Lorsqu'on a une machine avec une adresse IP variable, et qu'on veut la joindre depuis l'extérieur (pour se connecter à sa webcam, ou bien à un serveur HTTP qu'elle héberge ou bien pour tout autre raison), il est courant d'utiliser une mise à jour dynamique du DNS. Après tout, c'est bien le but principal du DNS <https://www.bortzmeyer.org/pourquoi-le-dns.html>, fournir un identificateur stable, le nom de domaine. Ce RFC explique les pièges et les problèmes lorsqu'on est connecté via un système à fort partage d'adresses IP (comme DS-Lite), et comment on peut utiliser PCP dans ce contexte.

Il existe plusieurs fournisseurs qui hébergent votre nom de domaine et acceptent des requêtes de mise à jour dynamiques, via un formulaire Web ou bien via une API. On peut avoir une idée du nombre de tels fournisseurs en regardant la liste de DNSlookup <http://dnslookup.me/dynamic-dns/>. Il n'y a pas à proprement parler de norme pour cette demande de mise à jour, à part le RFC 2136 [1] qui ne semble guère utilisé par ces fournisseurs (comme le note à juste titre le Wikipédia anglophone, « DNS dynamique » désigne en fait deux choses différentes, la mise à jour par le RFC 2136 et le fait de mettre à jour la base sans édition manuelle). Le fournisseur doit avoir une interface qui assure en outre un minimum de sécurité (cf. la section 4 de notre RFC) par exemple en utilisant HTTPS + une authentification du client. Une utilisation courante est que le routeur ou l'ordinateur de l'utilisateur détecte une nouvelle adresse IP, et il notifie l'hébergeur de cette nouvelle adresse. Ce dernier met alors à jour le DNS (via le RFC 2136 ou via toute autre méthode).

Mais, aujourd'hui, de plus en plus d'utilisateurs sont coincés derrière un système à partage d'adresses massif (cf. RFC 6888), où on n'a plus une seule adresse IP publique à la maison. Pour être joint de

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc2136.txt

l'extérieur, il faut communiquer non seulement son adresse IP mais également un port. Plus question de faire tourner son serveur Web sur le port 80, celui-ci ne peut pas être partagé. En outre, il faudra évidemment indiquer au réseau que le port en question devra être connecté au port sur lequel écoute le serveur de l'utilisateur (ce qui se fait en UPnP ou, plus récemment, en PCP, ce dernier étant normalisé dans le RFC 6887).

Les nombreux problèmes posés par le partage d'adresses sont bien connus (RFC 6269). Parmi eux :

— On ne peut plus compter sur les ports « bien connus » (comme 80 pour HTTP), il faut pouvoir indiquer aux clients un port explicite,
— Il faut un mécanisme pour configurer les connexions entrantes (justement ce que fait PCP),
— Il faut pouvoir détecter des changements d'adresses IP qui se passent très loin de l'utilisateur, dans l'AFTR de DS-Lite (RFC 6333) ou dans le routeur CGN pour NAT444 <`https://www.bortzmeyer.org/nats.html`>.

À noter que, pour prévenir le monde extérieur du couple {adresse, port} à contacter (problème connu sous le nom de « problème du rendez-vous »), on utilise souvent aujourd'hui des techniques non-standard, spécifiques à une application particulière (cela se fait souvent dans le monde du jeu en ligne) ou spécifiques à un protocole donné (SIP va utiliser un "*SIP proxy*", section 16 du RFC 3261).

Voyons donc les trois problèmes à résoudre et les solutions possibles (section 2 du RFC) :

— Pour indiquer le port, deux méthodes, utiliser des URI qui permettent d'inclure le port, comme le permet HTTP (`http://www.example.net:5318/foo/bar`), ou utiliser des enregistrement DNS SRV (RFC 2782). La seconde méthode est plus propre, car invisible à l'utilisateur (et elle offre d'autres possibilités), et elle n'oblige pas les applications à utiliser des URI. Malheureusement, elle est anormalement peu déployée dans les applications (par exemple les navigateurs Web).
— Pour configurer les connexions entrantes, on se sert de PCP (RFC 6887), dont c'est justement le but principal.
— Pour détecter les changements d'adresses, on va encore se servir de PCP. Une des solutions est de faire des requêtes PCP de type `MAP` régulièrement, demandant une correspondance {adresse externe -¿ adresse interne} de courte durée et de voir quelle est l'adresse IP retournée. Une autre solution, moins bavarde, est de juste vérifier la table locale du client PCP, notant les changements d'adresses, et de ne faire une requête `MAP` que s'il n'y a pas de correspondance dans la table locale. Dans les deux cas, lorsqu'on détecte un changement d'adresse, on met à jour le DNS par les moyens classiques (le client PCP, routeur ou ordinateur, est également client du service de DNS dynamique).

La section 3 couvre quelques détails pratiques de déploiement. Par exemple, le RFC couvre le cas où le service de « DNS dynamique » ne fait pas de DNS du tout mais publie sa propre adresse dans le DNS, redirigeant ensuite les clients vers le bon serveur (via une réponse 301 en HTTP, ce qui se nomme parfois "*URL forwarding*"). Ces serveurs devront modifier leur API pour que le client puisse indiquer le port en plus de l'adresse IP (aujourd'hui, ils ont souvent 80 câblé en dur). Dans un contexte différent (protocole ne permettant pas les redirections), on peut envisager d'utiliser la technique de découverte du RFC 6763.

Pour résumer par des remarques personnelles, tout cela est plutôt embrouillé, et on n'est pas tiré d'affaire, d'autant plus que PCP n'est pas tellement déployé aujourd'hui.