# RFC 8765 : DNS Push Notifications

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 23 juin 2020

Date de publication du RFC : Juin 2020

https://www.bortzmeyer.org/8765.html

---

Lorsqu'une donnée DNS change dans les serveurs faisant autorité, les clients ne seront prévenus que lorsqu'ils reviendront demander, et cela peut être long s'ils ont mémorisé l'ancienne valeur. Ce RFC propose une autre approche, où les données DNS sont **poussées** vers les clients au lieu d'attendre qu'ils tirent.

Qu'est-ce qui fait que des données DNS changent ? Cela peut être l'administrateur système qui a changé la zone et rechargé le serveur faisant autorité. Ou bien il y a pu y avoir mise à jour dynamique (RFC 2136 [1]). Ou encore d'autres mécanismes. Dans tous les cas, certains clients DNS (par exemple pour la découverte de services du RFC 6763) aimeraient bien être prévenus tout de suite (et voir la nouvelle imprimante du réseau local apparaitre dans la liste). Le DNS est purement "*pull*" et ces clients voudraient un peu de "*push*".

Cette idée de protocoles "*push*", où on envoie les données vers les clients au lieu d'attendre qu'il les réclame, est nouvelle pour le DNS mais ancienne dans l'Internet (modèle "*publish/subscribe*" ou "*design pattern*" Observateur). Parmi les protocoles IETF, XMPP peut fonctionner ainsi (cf. XEP0060 <https://xmpp.org/extensions/xep-0060.html>).

Il y a quand même un peu de "*push*" dans une variante du DNS, mDNS (RFC 6762, section 5.2), où les changements peuvent être poussés vers une adresse IP "*multicast*". Mais des protocoles comme la découverte de services du RFC 6763 peuvent aussi fonctionner sans "*multicast*" et cette solution ne leur convient donc pas. Il existait une solution non-standard, LLQ ("*Long-Lived Queries*"), décrite dans le RFC 8764 et mise en œuvre dans les produits Apple (cf. RFC 6281). Utilisant UDP, elle ne convient guère au DNS moderne.

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

Il existe aujourd'hui un mécanisme plus pratique pour déployer des fonctions comme le "*push*" : DSO ("*DNS Stateful Operations*", RFC 8490), qui repose sur des connexions de longue durée, sur TCP (et TLS pour la sécurité). La solution "*push*" est donc bâtie sur DSO, et il est donc bon d'avoir lu le RFC 8490 avant celui-ci.

La section 3 de notre RFC présente le protocole "*DNS push*". Un client s'abonne à un service de notification et, une fois abonné, il reçoit les nouveautés. Quand il n'a plus besoin des données, le client se déconnecte. L'abonnement est valable pour un certain couple {nom de domaine, type de données}. Client et serveur utilisent le mécanisme DSO du RFC 8490. Comment le client trouve-t-il le serveur ? Ce n'est pas forcément un des serveurs listés comme faisant autorité pour la zone. Il peut être trouvé en demandant à son résolveur normal (s'il accepte DSO, le client peut tenter sa chance) dans le DNS, puis, si ça ne marche pas, via une requête SRV pour le nom `_dns-push-tls._tcp.LA-ZONE` (cf. section 6, et son enregistrement à l'IANA <`https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml`>). Rappelez-vous que, si un serveur DNS accepte DSO mais pas "*DNS push*", il répondra avec un code de retour DNS <`https://www.iana.org/assignments/dns-parameters/dns-parameters.xml#dns-parameters-6`> `DSOTYPENI` (RFC 8490, section 5.1.1).

Pour éviter de trop charger le serveur, le client ne doit s'abonner que s'il a une bonne raison, par exemple parce qu'une fenêtre de sélection d'une imprimante est ouverte, et qu'il veut la rafraichir en permanence. Lorsque le client va s'endormir ou estomper son écran pour économiser sa batterie, il devrait se désabonner d'abord. En tout cas, le mécanisme "*DNS push*" ne devrait pas être utilisé 24x7, ce n'est pas son but. (Et c'est inutile, le mécanisme est suffisamment rapide pour pouvoir être invoqué seulement quand on en a besoin.)

De toute façon, le serveur n'est pas obligé d'accepter tous les clients qui se présentent. Il peut rejeter les abonnements s'il manque de ressources (section 4 du RFC).

Comment est-ce que le client demande du "*DNS push*" ? En s'abonnant, une fois la session DSO établie. Il envoie une requête de type DSO <`https://www.iana.org/assignments/dns-parameters/dns-parameters.xml#dns-dso-type-codes`> `SUBSCRIBE` (la syntaxe des TLV DSO est dans le RFC 8490, section 5.4.4). Dans les données de la requête se trouvent le nom de domaine et le type de données qui intéressent le client. Le serveur répond alors `NOERROR` (ou bien un code DNS d'erreur s'il y a un problème, par exemple `REFUSED` si le serveur ne veut pas) et envoie une réponse de type `SUBSCRIBE` (même type que la requête, c'est le bit QR de l'en-tête DNS qui permet de différencier requête et réponse). Il est parfaitement légitime pour un client de s'abonner à un nom de domaine qui n'existe pas encore, et le serveur ne doit donc jamais répondre `NXDOMAIN`.

Une fois l'abonnement accepté, le client attend tranquillement et, lorsqu'un changement survient, le serveur envoie des messages de type DSO `PUSH` (code 0x0041 <`https://www.iana.org/assignments/dns-parameters/dns-parameters.xml#dns-dso-type-codes`>). Les données contenues dans ce message sont formatées comme les messages DNS habituels, avec une exception : un TTL valant 0xFFFFFFFF signifie que l'ensemble d'enregistrements concerné a été supprimé.

À la fin, quand le client n'est plus intéressé, il envoie un message DSO `UNSUBSCRIBE` (code 0x0042). Encore plus radical, le client peut mettre fin à la session DSO, annulant ainsi tous ses abonnements.

Et la sécurité de ce "*DNS push*" (section 7 du RFC) ? "*DNS push*" impose l'utilisation de TLS (donc on a du DSO sur TLS sur TCP). Le client doit authentifier le serveur selon le profil strict du RFC 8310. Ensuite, DNSSEC est recommandé pour la découverte du serveur "*DNS push*", lorsqu'on utilise les requêtes SRV, puis DANE pour vérifier le serveur auquel on se connecte (RFC 7673).

Ensuite, "*DNS push*" permet d'utiliser les données précoces du RFC 8446 (section 2.3), ce qui permet de diminuer la latence <https://www.bortzmeyer.org/latence.html>. Mais elles posent quelques problèmes de sécurité : plus de confidentialité persistante, et risque de duplication des messages. D'autre part, si on utilise la reprise de session du RFC 8446 (section 2.2), il faut noter que, si elle permet de ne pas recommencer toute la session TLS, en revanche, elle ne garde pas les abonnements "*DNS push*", que le client devra donc renouveler.

Désolé, mais je ne connais pas encore de mise en œuvre de ce RFC, à part dans le monde Apple.