RFC 8094 : DNS over Datagram Transport Layer Security (DTLS)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 2 mars 2017

Date de publication du RFC : Février 2017

https://www.bortzmeyer.org/8094.html

Le DNS fonctionne traditionnellement surtout sur UDP, notamment pour minimiser la latence : quand on veut une réponse DNS, on la veut rapidement. Dans le cadre du projet « DNS et vie privée », le choix avait été fait de chiffrer le trafic DNS avec TLS (RFC 7858 [1]), imposant ainsi l'usage de TCP. Certains pensaient quand même qu'UDP était bien adapté au DNS et, puisqu'il existe une version de TLS adaptée à UDP, DTLS, ce serait une bonne idée de l'utiliser pour chiffrer le DNS. C'est ce que décrit ce nouveau RFC (qui ne semble pas avoir un avenir brillant, peu de gens sont intéressés).

De toute façon, il est très possible que le DNS utilise de plus en plus TCP, et le RFC 7766 allait dans ce sens, demandant davantage de la part des mises en œuvre de DNS sur TCP. Mais, bon, il est toujours bon d'essayer des alternatives, d'où ce RFC, dans l'état « Expérimental ». Outre les RFC déjà cités, il est recommandé, avant de le lire, de prendre connaissance du RFC 7626, qui décrit les problèmes de vie privée que pose le DNS, et le RFC 6347, qui normalise DTLS (bien moins connu que son copain TLS, et peu utilisé jusqu'à présent, à part pour WebRTC).

Les motivations pour explorer une alternative au DNS-sur-TLS du RFC 7858 sont :

— TCP souffre du « "*head of line blocking*" » où la perte d'un seul paquet empêche de recevoir tous ceux qui suivent, même s'ils sont bien arrivés, tant que le paquet perdu n'est pas retransmis. DNS-sur-DTLS sera donc peut-être meilleur sur des réseaux qui perdent pas mal de paquets.
— Dans certaines conditions, l'établissement d'une session est plus rapide avec DTLS qu'avec TLS. (Rappelez-vous toutefois que le RFC 7766 exige des sessions TCP persistentes : pas question d'établir une session par requête DNS!) Reprendre une session TLS peut ne prendre qu'un aller-retour avec DTLS, alors que TLS devra attendre l'établissement de la connexion TCP (le RFC 7413 changera peut-être les choses, mais TLS et DTLS 1.3 obligeront également à réviser ce raisonnement.)

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7858.txt

De même qu'un serveur et un client DNS ne peuvent pas se contenter d'UDP (pour pouvoir envoyer des données de grande taille, il faudra de toute façon passer à TCP), DNS-sur-DTLS ne peut pas suffire seul, et il faudra donc que les clients et serveurs aient également DNS-sur-TLS.

La spécification de DNS-sur-DTLS est dans la section 3 de notre RFC. DNS-sur-DTLS va tourner, comme DNS-sur-TLS, sur le port 853 (sauf accord préalable entre client et serveur, s'ils sont adultes et consentants). Un client peut déterminer si le serveur gère DNS-sur-DTLS en envoyant un message DTLS `ClientHello` vers le port 853. En l'absence de réponse, le client réessaie, puis laisse tomber DTLS. Selon sa configuration (plus ou moins paranoïaque), le client va alors tenter le DNS habituel en clair, ou bien complètement renoncer. En tout cas, interdiction d'utiliser le port 853 pour transmettre des messages DNS en clair. L'utilisation de ce port sur UDP implique DTLS.

Si, par contre, le serveur répond et qu'une session DTLS est établie, le client DNS-sur-DTLS authentifie le serveur avec les mêmes méthodes que pour TLS, en suivant les bonnes pratiques de sécurité de TLS (RFC 7525) et les profils d'authentification de DNS-sur-TLS décrits dans le RFC 8310. Une fois que tout cela est fait, les requêtes et réponses DNS sont protégées et les surveillants sont bien embêtés, ce qui était le but.

DTLS tourne sur UDP et reprend sa sémantique. Notamment, il est parfaitement normal qu'une réponse arrive avant une autre, même partie plus tôt. Le client DNS-sur-DTLS ne doit donc pas s'étonner et, pour faire correspondre les requêtes et les réponses, il doit, comme avec le DNS classique sur UDP, utiliser le "*Query ID*" ainsi que la question posée (qui est répétée dans les réponses, dans la section Question).

Pour ne pas écrouler le serveur sous la charge, le client ne devrait créer qu'une seule session DTLS vers chaque serveur auquel il parle, et y faire passer tous les paquets. S'il y a peu de requêtes, et que le client se demande si le serveur est toujours là, il peut utiliser l'extension TLS du « battement de cœur » (RFC 6520), qui peut également servir à rafraichir l'état d'un routeur NAT éventuel. Le RFC recommande aux serveurs DNS-sur-DTLS un délai d'au moins une seconde en cas d'inutilisation de la session, avant de raccrocher. Le problème est délicat : si ce délai est trop long, le serveur va garder des ressources inutiles, s'il est trop court, il obligera à refaire le travail d'établissement de session trop souvent. En tout cas, le client doit être prêt à ce que le serveur ait détruit la session unilatéralement, et doit la réétablir s'il reçoit l'alerte DTLS qui lui indique que sa session n'existe plus.

Un petit mot sur les performances, maintenant, puisque rappelons-nous que le DNS doit aller **vite** (section 4). L'établissement d'une session DTLS peut nécessiter d'envoyer des certificats, qui sont assez gros et peuvent nécessiter plusieurs paquets. Il peut donc être utile d'utiliser les clés brutes (pas de certificat) du RFC 7250, ou bien l'extension TLS "*Cached Information Extension*" (RFC 7924).

Dans le cas d'un lien "*stub resolver*" vers résolveur, le serveur DNS parle à beaucoup de clients, chaque client ne parle qu'à très peu de serveurs. L'état décrivant les sessions DTLS doit donc plutôt être gardé chez le client (RFC 5077). Cela permettra de réétablir les sessions DTLS rapidement, sans pour autant garder d'état sur le serveur.

Le DNS est la principale application qui se tape les problèmes de PMTU ("*Path MTU*", la MTU du chemin complet). Les réponses DNS peuvent dépasser les 1 500 octets magiques (la MTU d'Ethernet et, de facto, la PMTU de l'Internet). DTLS ajoute au moins 13 octets à chaque paquet, sans compter l'effet du chiffrement. Il est donc impératif (section 5) que clients et serveurs DNS-sur-DTLS gèrent EDNS (RFC 6891) pour ne pas être limité par l'ancien maximum DNS de 512 octets, et que les serveurs limitent les paquets DTLS à la PMTU (RFC 6347).

Contrairement au DNS classique, où chaque requête est indépendante, toute solution de cryptographie va nécessiter un état, l'ensemble des paramètres cryptographiques de la session. L'"*anycast*", qui est répandu pour le DNS, ne pose donc pas de problème au DNS classique : si le routage change d'avis entre deux requêtes, et que la seconde requête est envoyée à un autre serveur, aucun problème. Avec DTLS, ce n'est plus le cas (section 6 du RFC) : le deuxième serveur n'a pas en mémoire la session cryptographique utilisée. Le serveur qui la reçoit va répondre avec une alerte TLS fatale (la méthode recommandée) ou, pire, ne pas répondre. Dans les deux cas, le client doit détecter le problème et réétablir une session cryptographique. (À noter que l'alerte TLS n'est pas authentifiée et ne peut donc pas être utilisée comme seule indication du problème. C'est d'ailleurs pareil pour d'éventuels messages d'erreur ICMP.) Le cas est donc proche de celui où le serveur ferme la session unilatéralement, et la solution est la même : le client doit toujours être prêt à recommencer l'ouverture de session DTLS.

Un point de sécurité, pour finir (section 9). Le RFC recommande l'utilisation de l'extension TLS « agrafage OCSP » (RFC 6066, section 8), notamment pour éviter la grosse fuite d'information que représente OCSP.

Il n'existe aucune mise en œuvre de DNS-sur-DTLS, et aucune n'est prévue. L'avenir de cette expérimentation est... incertain, à moins qu'un[Caractère Unicode non montré [2]]e courageu[Caractère Unicode non montré]x[Caractère Unicode non montré]se développeu[Caractère Unicode non montré]r[Caractère Unicode non montré]se ne s'y mette ?

2. Car trop difficile à faire afficher par LaTeX