RFC 9953 : DNS over the Constrained Application Protocol (DoC)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 1 avril 2026

Date de publication du RFC : Mars 2026

https://www.bortzmeyer.org/9953.html

Le protocole CoAP est un protocole léger, conçu pour les objets connectés. Ce RFC décrit comment faire du DNS au-dessus de CoAP. Si votre brosse à dents a besoin de faire du DNS, c'est ce RFC qu'il faut lire.

CoAP, normalisé dans le RFC 7252[1], ressemble à HTTP mais en plus simple. Il vise le marché des objets contraints (en mémoire, énergie, en capacité réseau, en puissance de calcul…, voir le RFC 7228). CoAP permet à un client d'envoyer des requêtes à un serveur, et notre tout neuf RFC 9953 explique comment utiliser CoAP pour faire des requêtes DNS et recevoir des réponses DNS. Comme CoAP peut être vu, de manière simplifiée, comme une variante légère de HTTP, ce DNS sur CoAP, alias DoC ("*DNS over CoAP*"), est très proche dans ses principes de DoH (RFC 8484). Les contraintes de l'Internet des objets empêchent votre tournevis ou votre ampoule électrique de faire du DoH (qui implique HTTPS), d'où ce DoC. Bien sûr, il y avait toujours la possibilité de faire du DNS sur DTLS, comme le décrit le RFC 8094, mais CoAP a plein de fonctions sympas pour les objets contraints (transferts par blocs - RFC 7959, qui résout les problèmes de MTU, relais CoAP, etc).

L'architecture générale de DoC est très proche de celle de DoH : le client DoC interroge un serveur DoC qui est lui-même un client DNS, parlant typiquement à un résolveur DNS <https://www.bortzmeyer.org/resolveur-dns.html>. La communication entre le client DoC et le serveur DoC utilise CoAP, celle entre le serveur DoC et le résolveur utilise le DNS normal, sur UDP, TCP, TLS, etc.

Le serveur DoC sera a priori désigné par le chemin `/`. Le but est que les messages soient plus courts (DoH utilise souvent le chemin `/dns-query`). Au fait, comment le client DoC trouve-t-il le serveur (section 3 du RFC) ? Plusieurs méthodes sont admises, de la configuration manuelle (« le serveur DoC

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7252.txt

est en coaps://doc.foobar.example/ ») aux répertoires CORE (RFC 9176) en passant par les classiques DHCP ou RA ou bien par la découverte de résolveurs du RFC 9462. Ces différentes solutions sont plus ou moins sûres, et plus ou moins pratiques pour l'administrateur/utilisateur (et en général les plus pratiques sont les moins sûres).

Mais ce n'est pas tout : le serveur DoC peut aussi être trouvé via le DNS, avec un enregistrement SVCB (RFC 9460 et RFC 9461), si l'objet contraint a déjà un moyen d'interroger le DNS (la clé SVCB est docpath, numéro 10 <https://www.iana.org/assignments/dns-svcb/dns-svcb.xml#dns-svcparamkeys>).

Une fois qu'on connait le serveur, on peut lui parler. La section 4 décrit les messages CoAP échangés. Ils sont étiquetés avec le type application/dns-message (valeur numérique 553 <https://www.iana.org/assignments/core-parameters/core-parameters.xml#content-formats>).

La méthode CoAP utilisée est FETCH (RFC 8132), qui n'a pas d'équivalent HTTP, contrairement à la plupart des méthodes CoAP. DoC utilise le principe REST (RFC 6690). Le RFC recommande de ne pas oublier le champ Accept: pour indiquer le type de données accepté. Le type recommandé est application/dns-message. Dans la requête, le "*Query ID*" DNS devrait être à 0, CoAP se charge de mettre en correspondance requêtes et réponses donc cet identificateur n'est pas utile (c'est pareil pour DoH).

Et les réponses ? Là encore, comme pour DoH, les codes de retour CoAP indiquent si le serveur DoC a pu être joint et a pu faire son travail, **pas** si la requête DNS a eu une réponse satisfaisante. Autrement dit, tant que le serveur DoC fonctionne bien, il renvoie le code 2.05 (RFC 7252, section 5.9.1.5.) et c'est dans le message DNS qu'on saura si la résolution DNS a renvoyé NOERROR, SERVFAIL, NXDOMAIN, etc.

La mémorisation des réponses joue un rôle plus grand en CoAP qu'en HTTP (rappelez-vous que CoAP sert à des objets contraints). Le RFC demande donc que les TTL dans la réponse DNS soient constants, afin que l'ETag (RFC 7252, section 5.10.6), s'il est calculé via un condensat, ne change pas. Le serveur DoC doit par contre mettre un champ Max-Age: dans sa réponse (section 4.3.2).

Voici, tiré du RFC, un exemple de réponse CoAP, formaté en texte :

```
2.05 Content
Content-Format: 553 (application/dns-message)
Max-Age: 58719
Payload (human-readable):
    ;; ->>Header<<- opcode: QUERY, status: NOERROR, id: 0
    ;; flags: qr rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

    ;; QUESTION SECTION:
    ;example.org.                IN      AAAA
    ;; ANSWER SECTION:
    ;example.org.         79689   IN      AAAA    2001:db8:1:0:1:2:3:4
```

Et la sécurité ? Le RFC recommande de sécuriser la communication avec le système OSCORE du RFC 8613 (ou, sinon, le DTLS du RFC 9147, ce qui donne du CoAPS). Le RFC note que cela assure la sécurité de la communication entre le client DoC et le serveur DoC mais que, après, on ne sait pas ce que fait le serveur (la section 8 recommande évidemment que le serveur DoC valide avec DNSSEC).

Questions mises en œuvre de DoC, on trouve (mais je ne les ai pas testées, je ne fais pas d'Internet des Objets), par les auteurs du RFC, dans RIOT, un client <https://api.riot-os.org/group__net__gcoap__dns.html> en C (source sur Github <https://github.com/RIOT-OS/RIOT/tree/master/sys/net/application_layer/gcoap>) et un serveur <https://github.com/netd-tud/aiodnsprox> en Python.

Si vous voulez allez plus loin que ce résumé sommaire, une bonne lecture est le papier de 2023 des auteurs, « "*Securing Name Resolution in the IoT : DNS over CoAP*" <https://dl.acm.org/doi/10.1145/3609423> ».