RFC 9204 : QPACK: Header Compression for HTTP/3

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 7 juin 2022

Date de publication du RFC : Juin 2022

https://www.bortzmeyer.org/9204.html

QPACK, normalisé dans ce RFC est un mécanime de compression des en-têtes HTTP, prévu spécifiquement pour HTTP/3. Il est proche du HPACK de HTTP/2, mais adapté aux particularités de QUIC <https://www.bortzmeyer.org/quic.html>.

Car le HPACK du RFC 7541 [1] a un défaut qui n'était pas un problème en HTTP/2 mais le devient avec HTTP/3 : il supposait que les trames arrivent dans l'ordre d'émission, même si elles circulent sur des ruisseaux différents. Ce n'est plus vrai en HTTP/3 qui, grâce à son transport sous-jacent, QUIC <https://www.bortzmeyer.org/quic.html>, a davantage de parallélisme, et où une trame peut en doubler une autre (si elles n'étaient pas dans le même ruisseau). QPACK ressemble à HPACK, mais en ayant corrigé ce problème. (Au fait, ne cherchez pas ce que veut dire QPACK, ce n'est pas un acronyme.)

Donc, le principe de QPACK. Comme HPACK, on travaille avec deux tables, qui vont associer aux en-têtes HTTP un nombre (appelé index). L'une des tables est statique, définie dans ce RFC (annexe A) et donc identique pour tout le monde. L'autre est dynamique et construite par un échange de messages transmis dans des ruisseaux QUIC. Le fonctionnement avec la table statique est simple : l'encodeur regarde si ce qu'il veut écrire est dans la table, si oui, il le remplace par l'index. Le décodeur, recevant un index, le remplace par le contenu de la table. Par exemple, l'en-tête HTTP `if-none-match` (RFC 7232, section 3.2) est dans la table, index 9. L'encodeur remplacera donc un `if-none-match` par 9 (12 octets de gagnés si tout était sous forme de caractères de 8 bits, mais peut-être un peu plus ou un peu moins, avec l'encodage de QPACK), et le décodeur fera l'inverse.

J'ai un peu simplifié en supposant que la table ne contenait que les noms des en-têtes. Elle peut aussi contenir leur valeur si celle-ci est très courante. Ainsi, `accept: application/dns-message` est dans la table, index 30, vu son utilisation intensive par DoH (RFC 8484). Même chose pour `content-type: text/html;charset=utf-8` à l'index 52.

La table dynamique est évidemment bien plus complexe. Encodeur (celui qui comprime) et décodeur doivent cette fois partager un état. En outre, le parallélisme inhérent à QUIC fait qu'un message d'ajout d'une entrée dans la table pourrait arriver après le message utilisant cette entrée. QPACK fonctionne de la façon suivante :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7541.txt

— L'encodeur peut envoyer (sur un ruisseau QUIC dédié) des instructions de construction de la table dynamique, notamment pour insérer une nouvelle entrée (mais pas en supprimer).
— Le décodeur accuse réception de ces instructions.
— L'encodeur envoie l'index le plus élevé de sa table, ce qui permet au décodeur, en regardant sa propre table, de savoir s'il est synchronisé ou pas. S'il ne l'est pas, le décodeur se bloque en attendant les instructions d'insertion qui vont faire grandir sa table.

QUIC a un système de contrôle de flux, et des inconvénients peuvent en résulter, par exemple un blocage des messages de contrôle de la table. Pour éviter tout blocage, un encodeur peut n'utiliser que des entrées de la table qui ont déjà fait l'objet d'un accusé de réception. Il comprimera moins mais ne risquera pas d'être bloqué.

Le RFC détaille les précautions à prendre pour éviter l'interblocage. Ainsi, les messages modifiant la table risquent d'être bloqués par ce système alors que le récepteur n'autorise pas de nouvelles trames tant qu'il n'a pas traité des trames qui ont besoin de ces nouvelles entrées dans la table dynamique. L'encodeur a donc pour consigne de ne pas tenter de modifier la table s'il ne lui reste plus beaucoup de « crédits » d'envoi de données. (D'une manière générale, quand il y a des choses compliquées à faire, QPACK demande à l'encodeur de les faire, le décodeur restant plus simple.)

L'encodage des messages QPACK est spécifié dans la section 4. QPACK utilise deux ruisseaux QUIC unidirectionnels, un dans chaque direction. Ils sont enregistrés à l'IANA <https://www.iana.org/assignments/http3-parameters/http3-parameters.xml#http3-parameters-stream-types>.

Notez aussi qu'il y a deux instructions d'ajout d'une entrée dans la table dynamique, une qui ajoute une valeur litérale (comme « ajoute `accept-language: fr` ») et une qui ajoute une valeur exprimée sous forme d'une référence à une entrée d'une table (qui peut être la statique ou la dynamique). Par exemple, comme `accept-language` est dans la table statique, à l'index 72, on peut dire simplement « ajoute 72 : fr ». Encore quelques octets gagnés.

Dans la trame `SETTINGS` de HTTP/3, deux paramètres concernent spécialement QPACK, pour indiquer la taille maximale de la table dynamique, et le nombre maximal de ruisseaux bloqués. Ils sont placés dans un registre IANA <https://www.iana.org/assignments/http3-parameters/http3-parameters.xml#http3-parameters-settings>.

Quelques mots sur la sécurité : dans certaines conditions, un observateur peut obtenir des informations sur l'état des tables (cf. l'attaque CRIME) même s'il ne peut déchiffrer les données protégées par TLS, celui-ci ne masquant pas la taille. Bien sûr, on pourrait remplir avec des données bidons mais cela annulerait l'avantage de la compression. La section 7 du RFC donne quelques idées sur des mécanismes de limitation du risque.

L'annexe A du RFC spécifie la table statique et ses 98 entrées. Elle a été composée à partir de l'analyse de trafic HTTP en 2018. L'ordre des entrées n'est pas arbitraire : vu comment sont représentés les entiers, donc les index, dans QPACK, les entrées les plus fréquentes sont en premier, car QPACK utilise moins de bits pour les nombres les plus petits. Notez aussi que cette table comprend les en-têtes HTTP « classiques », comme `content-length` ou `set-cookie` mais aussi ce que HTTP/2 appelle les « pseudo-en-têtes », qui commencent par deux-points. C'est par exemple le cas de la méthode HTTP (`GET`, `PUT`, etc), notée `:method` ou, dans les réponses, du code de retour, noté `:status` (tiens, la table statique a une entrée pour le code 403 mais pas pour le 404).

Si vous envisagez de programmer QPACK, l'annexe B contient des exemples de dialogue entre encodeur et décodeur, et l'annexe C du pseudo-code pour l'encodeur.