

# RFC 9221 : An Unreliable Datagram Extension to QUIC

Stéphane Bortzmeyer  
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 1 avril 2022

Date de publication du RFC : Mars 2022

<https://www.bortzmeyer.org/9221.html>

---

Le protocole QUIC <<https://www.bortzmeyer.org/quic.html>> fournit, comme son concurrent TCP, un canal fiable d'envoi de données. Mais certaines applications n'ont pas impérativement besoin de cette fiabilité et se contenteraient très bien d'un service de type datagramme. Ce nouveau RFC ajoute donc un nouveau type de trame QUIC, DATAGRAM, pour fournir un tel service.

Dit comme cela, évidemment, ça semble drôle, de bâtir un service non fiable au-dessus du service fiable que rend QUIC. Mais cela permet plusieurs choses très intéressantes, notamment si deux machines qui communiquent ont besoin des deux types de service : elles pourront utiliser une seule session QUIC pour les deux.

Un petit rappel (mais relisez le RFC 9000<sup>1</sup> pour les détails) : dans une connexion QUIC sont envoyés des paquets et chaque paquet QUIC contient une ou plusieurs trames. Chaque trame a un type et ce type indique si la trame sera retransmise ou non en cas de perte. Le service habituel (utilisé par exemple par HTTP/3) se sert de trames de type `STREAM`, qui fournissent un service fiable (les données perdues sont retransmises par QUIC). Mais toutes les applications ne veulent pas d'une telle fiabilité ou, plus précisément, ne sont pas prêtes à en payer le coût en termes de performance. Ces applications sont celles qui préféreraient utiliser UDP (RFC 768) plutôt que TCP. Pour rajouter de la sécurité, elles utilisaient DTLS (RFC 6347). Si QUIC <<https://www.bortzmeyer.org/quic.html>> remplace TCP, comment remplacer UDP+DTLS?

D'ailleurs, faut-il le remplacer ? La section 2 du RFC donne les raisons suivantes :

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc9000.txt>

- Les applications qui ont besoin à la fois d'un canal fiable et d'un canal qui ne l'est pas (par exemple une application de communication qui veut un canal fiable pour la signalisation et un pas forcément fiable pour les données multimédia) peuvent avec QUIC n'avoir qu'une seule session (et donc ne « payer » l'établissement de la session qu'une fois, ce qui diminuera la latence <<https://www.bortzmeyer.org/latence.html>>),
- QUIC a un meilleur mécanisme de récupération des pertes que DTLS, lors de l'établissement de la connexion,
- contrairement à UDP+DTLS, QUIC a un mécanisme de contrôle de la congestion, ce qui simplifie la tâche de l'application (cf. RFC 8085).

Ces raisons sont particulièrement importantes pour le "*streaming*" audio/vidéo, pour les jeux en ligne, etc. Les datagrammes dans QUIC peuvent aussi être utiles pour un service de VPN. Là aussi, le VPN a besoin à la fois d'un canal fiable pour la configuration de la communication, mais peut se satisfaire d'un service de type datagramme ensuite (puisque les machines connectées via le VPN auront leur propre mécanisme de récupération des données perdues). Permettre la création de VPN au-dessus de QUIC est le projet du groupe de travail MASQUE <<https://datatracker.ietf.org/wg/masque/>>.

Bien, normalement, maintenant, vous êtes convaincu-es de l'intérêt des datagrammes au-dessus de QUIC. Pour en envoyer, il faut toutefois que les deux parties qui communiquent soient d'accord. C'est le rôle du paramètre de transport QUIC `max_datagram_frame_size`, à utiliser lors de l'établissement de la session. (Ce paramètre est enregistré à l'IANA <<https://www.iana.org/assignments/quic/quic.xml#quic-transport>>.) S'il est absent, on ne peut pas envoyer de datagrammes (et le partenaire coupe la connexion si on le fait quand même). S'il est présent, il indique la taille maximale acceptée, typiquement 65 535 octets. Sa valeur peut être stockée dans la mémoire d'une machine, de façon à permettre son utilisation lors du 0-RTT (commencer une session QUIC directement sans perdre du temps en négociations). Un paquet QUIC 0-RTT peut donc inclure des trames de type DATAGRAM.

Ah, justement, ce type DATAGRAM. Désormais enregistré à l'IANA <<https://www.iana.org/assignments/quic/quic.xml#quic-frame-types>> parmi les types de trames QUIC, il sert à indiquer des trames qui seront traitées comme des datagrammes, et qui ont la forme suivant :

```
DATAGRAM Frame {
    Type (i) = 0x30..0x31,
    [Length (i)],
    Datagram Data (..),
}
```

Le champ de longueur est optionnel, sa présence est indiquée par le dernier bit du type (0x30, pas de champ Longueur, 0x31, il y en a un). S'il est absent, la trame va jusqu'au bout du paquet QUIC (rappelons qu'un paquet QUIC peut contenir plusieurs trames).

Maintenant qu'on a des datagrammes comme QUIC, comment les utilise-t-on (section 5 du RFC)? Rien d'extraordinaire, l'application envoie une trame DATAGRAM et l'application à l'autre bout la recevra. Si la trame n'est pas arrivée, l'émetteur ne le saura pas. La trame peut se retrouver avec d'autres trames (du même type ou pas) dans un même paquet QUIC (et plusieurs paquets QUIC peuvent se retrouver dans un même datagramme IP). La notion de ruisseau ("*stream*") n'existe pas pour les trames-datagrammes, si on veut pouvoir les démultiplexer, c'est à l'application de se débrouiller. (Une version préliminaire de cette extension à QUIC prévoyait un mécanisme de démultiplexage, finalement abandonné.) Comme toutes les trames QUIC, les trames DATAGRAM sont protégées par la cryptographie. Le service est donc équivalent à celui d'UDP+DTLS, pas UDP seul.

Un service de datagrammes est non fiable, des données peuvent se perdre. Mais si les trames de type DATAGRAM ne sont pas réémises (pas par QUIC, en tout cas), elles entraînent néanmoins l'émission

d'accusés de réception (RFC 9002, sections 2 et 3). Une bibliothèque QUIC peut ainsi (mais ce n'est pas obligatoire) notifier l'application des pertes. De la même façon, elle a la possibilité de dire à l'application quelles données ont été reçues par le QUIC à l'autre bout (ce qui ne garantit pas que l'application à l'autre bout les aient bien reçues, si on a besoin d'informations sûres, il faut le faire au niveau applicatif).

Comme l'émetteur n'estime pas crucial que toutes les données arrivent, une optimisation possible pour QUIC est que l'accusé de réception d'un paquet ne contenant que des trames DATAGRAM ne soit pas émis tout de suite, il peut attendre le paquet suivant.

Cette extension existe déjà dans plusieurs mises en œuvre de QUIC, mais je ne l'ai pas testée.