RFC 9002 : QUIC Loss Detection and Congestion Control

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 mai 2021

Date de publication du RFC : Mai 2021

https://www.bortzmeyer.org/9002.html

Pour tout protocole de transport, détecter les pertes de paquets, et être capable d'émettre et de réémettre des paquets sans provoquer de congestion sont deux tâches essentielles. Ce RFC explique comment le protocole QUIC <https://www.bortzmeyer.org/quic.html> assure cette tâche.

Pour l'instant, TCP reste le principal protocole de transport sur l'Internet. Mais QUIC <https://www.bortzmeyer.org/quic.html> pourrait le dépasser. QUIC est normalisé dans une série de RFC et notre RFC 9002 [1] se charge d'une tâche délicate et cruciale : expliquer comment détecter les pertes de paquets, et comment ne pas contribuer à la congestion. Voyons d'abord la conception générale (section 3 du RFC). Les messages QUIC sont mis dans des trames, une ou plusieurs trames sont regroupées dans un paquet (qui n'est pas un paquet IP) et un ou plusieurs paquets sont dans un datagramme UDP qu'on envoie à son correspondant. Les paquets ont un numéro (RFC 9000, section 12.3). Ces numéros ne sont pas des numéros des octets dans les données envoyées, notamment, un numéro de paquet ne se répète jamais dans une connexion. Alors qu'on peut envoyer les mêmes données plusieurs fois, s'il y a perte et réémission ; en cas de retransmission, les données sont renvoyées dans un nouveau paquet, avec un nouveau numéro, contrairement à TCP. Cela permet de savoir facilement si c'est une retransmission. (TCP, lui, essaie de déduire l'ordre de distribution du numéro de séquence, et ce n'est pas trivial.)

La plupart des paquets QUIC feront l'objet d'un accusé de réception mais attention. Il y a des trames dont le type exige un accusé de réception et d'autres non. Si un paquet ne contient que des trames n'exigeant pas d'accusé de réception, ce paquet ne sera confirmé par le récepteur qu'indirectement, lors de la réception d'un paquet ultérieur contenant au moins une trame exigeant un accusé de réception.

QUIC n'est pas TCP, cela vaut la peine de le rappeler. La très intéressante section 4 du RFC enfonce le clou en énumérant les différences entre les algorithmes de TCP et ceux de QUIC, pour assurer les mêmes

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc9002.txt

fonctions. Ainsi, dans TCP, tous les octets sont numérotés selon un seul espace de numérotation (les numéros de séquence) alors que QUIC a plusieurs espaces, les paquets servant à établir la connexion ne partagent pas leurs numéros avec ceux des données. QUIC fonctionne ainsi car les premiers sont moins protégés par la cryptographie.

Pour TCP, le numéro de séquence indique à la fois l'ordre d'émission et l'ordre de l'octet dans le flux de données. Le problème de cette approche est que, en cas de retransmission, le numéro de séquence n'indique plus l'ordre d'émission, rendant difficile de distinguer une émission initiale et une retransmission (ce qui serait pourtant bien utile pour estimer le RTT). Au contraire, dans QUIC, le numéro de paquet n'identifie que l'ordre d'émission. La retransmission a donc forcément un numéro supérieur à l'émission initiale. Pour déterminer la place des octets dans le flux de données, afin de s'assurer que l'application reçoive les données dans l'ordre, QUIC utilise le champ "*Offset*" des trames de type `STREAM`, celles qui transmettent les données (RFC 9000, section 19.8). QUIC a ainsi moins d'ambiguités, par exemple quand il faut mesurer le taux de pertes.

QUIC, comme TCP, doit estimer le temps optimum pour décider qu'un paquet est perdu (RTO, pour "*Retransmission TimeOut*"). QUIC est plus proche de l'algorithme du RFC 8985 que du TCP classique. La section 5 du RFC détaille l'estimation du RTT.

La section 6 porte sur le problème délicat de la détection des pertes de paquets. La plupart des paquets QUIC doivent faire l'objet d'un accusé de réception. S'il n'est pas arrivé avant un temps limite, le paquet est décrété perdu, et il faudra demander une réémission (RFC 9000, section 13.3). Plus précisement, le paquet est considéré comme perdu s'il avait été envoyé avant un paquet qui a fait l'objet d'un accusé de réception et s'il s'est écoulé N paquets depuis ou bien un temps suffisamment long. (TCP fait face à exactement le même défi, et la lecture des RFC 5681, RFC 5827, RFC 6675 et RFC 8985 est recommandée.) La valeur recommandée pour N est 3, pour être proche de TCP. Mais attention si le réseau fait que les paquets arrivent souvent dans le désordre, cela pourrait mener à des paquets considérés à tort comme perdus. Le problème existait déjà pour TCP mais il est pire avec QUIC puisque des équipements intermédiaires sur le réseau qui remettaient les paquets dans l'ordre ne peuvent plus fonctionner avec QUIC, qui chiffre le plus de choses possibles pour éviter ces interventions souvent maladroites. Et le délai avant lequel on déclare qu'un paquet est perdu ? Il doit tenir compte du RTT qu'on doit donc mesurer.

Une fois la ou les pertes détectées, on réémet les paquets. Simple, non ? Sauf qu'il faut éviter que cette réémission n'aggrave les problèmes et ne mène à la congestion (le réseau, trop chargé, perd des paquets, les émetteurs réémettent, chargeant le réseau, donc on perd davantage de paquets, donc les émetteurs réémettent encore plus…). L'algorithme actuellement spécifié pour QUIC (section 7 du RFC) est proche du NewReno de TCP (normalisé dans le RFC 6582). Mais le choix d'un algorithme de contrôle de l'émetteur est unilatéral, et une mise en œuvre de QUIC peut toujours en choisir un autre, comme Cubic (RFC 8312). Évidemment, pas question d'être le gros porc qui s'attribue toute la capacité `<https://www.bortzmeyer.org/capacite.html>` réseau pour lui seul, et cet algorithme doit de toute façon respecter les principes du RFC 8085 (en résumé : ne soyez pas égoïste, et pensez aux autres, laissez-leur de la capacité).

Pour aider à cette lutte contre la congestion, QUIC, comme TCP, peut utiliser ECN (RFC 3168 et RFC 8311).

Comme TCP, QUIC doit démarrer une nouvelle session doucement (RFC 6928) et non partir bille en tête avec une fenêtre de grande taille.

La réaction aux pertes de paquets peut avoir des conséquences sur la sécurité (section 8 du RFC). Par exemple, les « signaux » utilisés par QUIC pour décider qu'il y a eu une perte (l'absence d'un paquet,

le RTT, ECN) ne sont pas protégés par la cryptographie, contrairement aux données transportées et à certaines métadonnées. Un attaquant actif peut fausser ces signaux et mener QUIC à réduire son débit. Il n'y a pas vraiment de protection contre cela. Autre risque de sécurité, alors que QUIC est normalement conçu pour priver un observateur de beaucoup d'informations qui, avec TCP étaient en clair, il n'atteint pas 100 % de succès dans ce domaine. Par exemple les paquets ne contenant que des accusés de réception (trames de type `ACK`) peuvent être identifiés par leur taille (ils sont tout petits), et l'observateur peut alors en déduire des informations sur les performances du chemin. Si on veut éviter cela, il faut utiliser le remplissage des accusés de réception.

Vous aimez lire des programmes ? L'annexe A du RFC contient du pseudo-code mettant en œuvre les mécanismes de récupération décrits dans le RFC.