

RFC 6582 : The NewReno Modification to TCP's Fast Recovery Algorithm

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 7 avril 2012

Date de publication du RFC : Avril 2012

<https://www.bortzmeyer.org/6582.html>

Même les plus vieux protocoles, comme TCP, continuent à évoluer, dans l'environnement toujours changeant qu'est l'Internet. Pour lutter contre le principal ennemi d'un protocole de transport, la congestion, de nouveaux algorithmes sont régulièrement mis au point. Même des années après leur spécification, on trouve des choses à corriger ou améliorer. Ainsi, l'algorithme NewReno, originellement proposé par le RFC 2582¹ en 1999, puis normalisé par le RFC 3782 en 2004, voit avec ce nouveau RFC sa description évoluer très légèrement, pour répondre aux problèmes rencontrés.

La référence pour la lutte de TCP contre la congestion est le RFC 5681. Ce dernier spécifie quatre algorithmes possibles, et autorise des modifications limitées à ces algorithmes, par exemple en utilisant les accusés de réception sélectifs (SACK, RFC 2883) ou bien en utilisant les accusés de réception partiels (ceux qui ne couvrent pas la totalité des données encore en transit). C'est le cas de NewReno et ce RFC remplace l'ancienne définition de NewReno (celle du RFC 3782).

Un peu d'histoire de NewReno (section 1) : l'algorithme de retransmission rapide ("*Fast Recovery*") de TCP avait d'abord été mis en œuvre dans une version de BSD connue sous le nom de Reno. (L'article original <<ftp://ftp.ee.lbl.gov/papers/fastretrans.ps>> est toujours en ligne, ainsi qu'une comparaison des méthodes <<ftp://ftp.ee.lbl.gov/papers/sacks.ps.z>>.) L'idée de base de la retransmission rapide est de retransmettre les accusés de réception lorsqu'arrivent des paquets dont les numéros de séquence sont supérieurs à ceux attendus (section 5 de notre RFC; cette retransmission indique la perte de paquets intermédiaires). L'émetteur commence alors tout de suite à retransmettre, sans attendre l'expiration du délai de garde normal. Le problème est que l'émetteur ne sait pas exactement ce qu'il faut retransmettre. Par exemple, avec une MSS de 1460 octets et un numéro de séquence

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc2582.txt>

initial (ISN) de 67891, si un émetteur envoie cinq paquets, il sera satisfait lorsqu'il recevra un accusé de réception pour l'octet 75191. S'il reçoit des accusés de réception dupliqués pour l'octet 70811, cela veut dire que deux paquets sont arrivés, puis que le paquet n° 4 ou le n° 5 est arrivé, déclenchant l'émission des accusés de réception dupliqués. L'émetteur va alors retransmettre le paquet n° 3. Mais doit-il aussi renvoyer les n° 4 et n° 5? L'émetteur ne le sait pas, les accusés de réception n'indiquent pas les octets reçus hors-séquence, sauf à utiliser les SACK du RFC 2883. NewReno avait justement été conçu pour ce cas où on ne peut pas ou bien où on ne veut pas utiliser les accusés de réception sélectifs (SACK). Son principe est, pendant la retransmission rapide, de regarder si les accusés de réception couvrent tous les paquets transmis (accusé de réception pour l'octet 75191) ou bien seulement une partie (par exemple, accusé de réception pour l'octet 72271). Dans ce cas (**accusés de réception partiels**), on sait qu'il y a perte de plusieurs paquets et on peut retransmettre tout de suite (ici, le paquet n° 4).

NewReno est donc une légère modification, située uniquement chez l'émetteur, à l'algorithme de retransmission rapide (« OldReno »?). Cette modification est déclenchée par la réception, une fois la retransmission rapide commencée, d'accusés de réception partiels. Elle consiste en la réémission de paquets supplémentaires, suivant le paquet qu'« OldReno » retransmettait déjà.

Si vous voulez tous les détails, la section 3 précise le nouvel algorithme. Comme il ne nécessite des modifications que chez l'émetteur des données, il n'affecte pas l'interopérabilité des mises en œuvre de TCP. Évidemment, SACK serait une meilleure solution, donnant à l'émetteur toute l'information nécessaire mais, en son absence, NewReno améliore les performances en évitant d'attendre inutilement l'expiration d'un délai de garde.

Il existe quelques variations à NewReno, moins importantes que la différence entre OldReno et NewReno. Ces variations possibles sont documentées dans l'ancien RFC, le RFC 3782 et citées dans l'annexe A.

NewReno est aujourd'hui bien établi et éprouvé. L'ancien RFC, le RFC 3782 contenait des éléments en sa faveur (notamment via des simulations), qui semblent inutiles aujourd'hui et ont donc été retirés.

Globalement, les principaux changements depuis le RFC 3782 sont le retrait de nombreuses parties qui n'apparaissent plus aussi importantes qu'à l'époque. Le lecteur qui veut approfondir le sujet, et pas uniquement programmer NewReno correctement, peut donc lire le RFC 3782 pour découvrir d'autres discussions. Autrement, les changements sont de détail, et résumés dans l'annexe B.

Merci à Didier Barvaux pour sa relecture.