RFC 6069 : Making TCP more Robust to Long Connectivity Disruptions (TCP-LCD)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 13 décembre 2010

Date de publication du RFC : Décembre 2010

https://www.bortzmeyer.org/6069.html

Depuis très longtemps, un des problèmes du protocole de transport TCP est le fait qu'il réduise son débit en cas de pertes de paquets, pensant que cette perte provient d'une congestion. Mais si la perte de paquets était due à une coupure temporaire de la connectivité (ce qui est fréquent avec les liaisons radio), il est inutile de diminuer le débit, puisque le problème ne venait pas d'une congestion. TCP devrait au contraire continuer au même rythme, surtout une fois que la connectivité est rétablie. Une solution au problème pourrait être d'utiliser les messages ICMP pour mieux diagnostiquer la cause de la perte de paquets et c'est cette approche que ce RFC 6069 [1] suggère d'essayer.

Rappelons que TCP fonctionne en envoyant des données dont l'autre pair doit accuser réception. S'il ne le fait pas à temps, TCP réemet les données et, s'il n'y a toujours rien, TCP en déduit que le réseau est surchargé (et jette donc des paquets) et, bon citoyen, il réagit en diminuant le rythme d'envoi (RFC 2988 et section 1 de notre RFC 6069). C'est ce comportement qui évite à l'Internet de s'écrouler sous la charge. Il n'est donc pas question que TCP soit modifié pour envoyer systématiquement au maximum de ses possibilités, « au cas où ça passe ». Mais, parfois, TCP est excessivement prudent. Si je débranche puis rebranche un câble réseau pendant le transfert d'un gros fichier, TCP va ralentir alors qu'il n'y avait aucune congestion (voir l'article de Schuetz, S., Eggert, L., Schmid, S., et M. Brunner, « "*Protocol enhancements for intermittently connected hosts*" », dans "*SIGCOMM Computer Communication Review*" vol. 35, no. 3). En pratique, le cas le plus embêtant se produit avec les réseaux sans-fil où de tels « branchements/débranchements » sont fréquents, soit en raison d'une modification soudaine du médium (une pluie intense, si on est dehors, ou bien un parasite soudain), soit en raison d'un déplacement de l'ordinateur. Peut-on détecter qu'une absence d'accusés de réception était due à une coupure temporaire du réseau ?

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6069.txt

D'abord (section 2), il faut distinguer deux sortes de coupure du réseau. Les courtes sont celles qui durent moins longtemps que le délai de retransmission de TCP. Si les files d'attente du routeur situé avant la coupure sont suffisamment grandes, il pourra « tamponner » les paquets et il n'y aura même pas de perte, juste des variations de délai (cf. RFC 3522, RFC 4015 ou RFC 5682). Les longues coupures sont celles qui durent plus longtemps et où l'émetteur TCP doit donc commencer à renvoyer des paquets déjà transmis. Suivant le RFC 5681, TCP ne se contente pas de réémettre, il diminue son rythme de transmission. Si la connectivité revient, TCP ne va pas s'en apercevoir tout de suite et, même si les accusés de réception réapparaissent, TCP continuera à envoyer à un rythme réduit, sans bonne raison, juste parce qu'il a cru à la congestion. Idéalement, TCP devrait, au contraire, recommencer à pleine vitesse dès que la liaison est rétablie.

Comment détecter la coupure et le rétablissement ? La section 3 rappelle l'existence des paquets ICMP "*Destination Unreachable*" (RFC 792, RFC 1812 et RFC 4443). Ces paquets sont envoyés par le routeur, vers l'émetteur, si le routeur est obligé de jeter le paquet (codes 1, "*Host unreachable*" ou 0, "*Network unreachable*"). Mais attention, ils ne sont pas parfaits : ils ne sont pas envoyés en temps-réel (et donc arriveront peut-être après que TCP ait trouvé tout seul qu'il y a un problème) et ils sont en général limités en quantité.

Ces paquets ICMP contiennent les premiers octets du paquet qui a déclenché le problème et l'émetteur peut donc, en recevant le paquet ICMP, trouver la connexion TCP en cause. Le principe du nouvel algorithme expérimental TCP-LD (TCP "*Long Disruption*") est donc d'utiliser ces messages ICMP pour différencier la congestion et la coupure. Dans le cas d'une coupure, cela vaudra la peine de réessayer de manière plus agressive.

L'algorithme est présenté en section 4. Le principe est simple : lorsque TCP a déjà dépassé son délai de garde, attendant un accusé de réception, et en cas de réception d'un message ICMP indiquant une coupure, TCP ne va pas augmenter les délais de retransmission. Au retour des accusés de réception, TCP reprendra « plein pot ». Les détails figurent par la suite. D'abord, TCP-LD ne s'applique qu'aux TCP qui suivaient l'algorithme de retransmission du RFC 2988. L'algorithme ne doit être utilisé qu'une fois la connexion complètement établie. Les seuls messages ICMP pris en compte sont ceux qui sont émis en réponse à des données TCP qui ont fait l'objet d'une retransmission. (Et autres détails, l'algorithme complet figure dans cette section 4.)

Une discussion des différents points à garder en tête figure en section 5. Elle insiste sur le fait que l'algorithme TCP-LD n'est déclenché que s'il y a réception des messages ICMP indiquant une erreur **et** expiration du délai de garde. Cela garantit que TCP-LD ne sera utilisé qu'en cas de longue coupure. Il y a quand même des cas qui prennent TCP-LD en défaut. C'est le cas par exemple de l'ambiguité analysée en section 5.1. Cette ambiguité vient du fait que le paquet TCP (et donc le message ICMP d'erreur qui concerne ce paquet) n'indique pas s'il s'agit d'une transmission ou d'une retransmission. Ce n'est pas un problème en pratique mais, si l'émetteur TCP qui reçoit le paquet ICMP peut être sûr qu'il y a eu retransmission, il n'est pas forcément sûr que le message d'erreur était en réponse à la retransmission. Encore plus rigolo (section 5.2), TCP-LD peut associer à tort un message ICMP à une session TCP ayant connu une retransmission, simplement parce que le numéro de séquence a dépassé sa valeur maximale et est revenu au début (sur des réseaux rapides, les 32 bits qui stockent le numéro de séquence TCP ne suffisent pas longtemps). La probabilité que cela arrive **et** interfère avec une coupure réelle est toutefois très faible. D'autres problèmes amusants (pour ceux qui connaissent bien TCP) forment la fin de la section 4, comme par exemple les risques liés aux paquets dupliqués. Un certain nombre des problèmes exposés pourraient être résolus avec l'option d'estampillage temporel de TCP (RFC 7323 et section 6 de notre RFC) : les estampilles pourraient lever certaines ambiguités.

TCP-LD est une technique « côté émetteur » seulement, et qui peut donc être déployée unilatéralement. Y a-t-il des risques à le faire ? La section 7 explore ces risques. Par exemple (section 7.1), si un émetteur

TCP essaie de détecter une coupure définitive en se donnant une limite maximale au nombre de retransmissions, l'utilisation de TCP-LD, qui va réduire l'intervalle entre les retransmissions, pourra amener à des conclusions erronées. Il faut donc évaluer la durée maximale qu'on accepte avant de couper en temps, et pas en nombre de retransmissions.

Je l'ai dit au début, ce problème est perçu depuis très longtemps par les utilisateur de TCP. Il y a donc eu bien d'autres efforts pour le résoudre. La section 8 les résume. On y trouve par exemple des modifications des couches inférieures (RFC 3819) ou bien des modifications des routeurs IP, pour qu'ils analysent suffisamment de TCP pour pouvoir générer des messages d'information. TCP-LD, par contre, ne nécessite pas de modification des couches basses, ni des routeurs intermédiaires.

Un point important de ce schéma est que les messages ICMP ne sont sécurisés en rien et que d'autres RFC, comme le RFC 5927, demandent de s'en méfier et au minimum de ne pas agir sans les avoir sérieusement validés. Quels sont donc les risques (section 9)? Un attaquant pourrait, soit générer des faux messages ICMP "*Destination Unreachable*" pour que TCP, croyant à une coupure et pas à une congestion, continue à inonder le réseau, soit au contraire empêcher les messages ICMP d'arriver, ramenant TCP à la situation avant TCP-LD. Dans le premier cas, les paquets ICMP nécessiteraient d'inclure une partie du paquet TCP, dont des éléments difficiles à deviner, comme le numéro de séquence et les mécanismes de validation du RFC 5927 conviendraient donc. Si elles échouent, l'attaquant qui arrive à trouver le numéro de séquence et les autres informations a de toute façon la possibilité de monter des attaques bien pires.

L'algorithme TCP-LD avait été présenté en 2009 à la réunion IETF 75 de Stockholm. Les transparents (≪ "*Make TCP more Robust to Long Connectivity Disruptions*" ≫) sont en ligne `<http://www.ietf.org/proceedings/75/slides/tcpm-0.pdf>`. Ils exposent l'algorithme et le résultat d'une évaluation après mise en œuvre sur Linux. Plein de jolis graphes. À propos de Linux, le travail des auteurs était disponible en `<http://www.umic-mesh.net/downloads/linux-tcp.html>` et a depuis été intégré (sous une forme modifiée) dans le noyau standard : regardez `tcp_v4_err()` dans `net/ipv4/tcp_ipv4.c` et les commentaires mentionnant `draft-zimmermann-tcp-lcd` (l'ancien nom de ce RFC 6069).