RFC 5461 : TCP's Reaction to Soft Errors

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 26 février 2009

Date de publication du RFC : Février 2009

https://www.bortzmeyer.org/5461.html

Que doit faire une implémentation de TCP lorsqu'une erreur est signalée par le réseau, par exemple lorsqu'un paquet ICMP indique une route inaccessible ? La norme traditionnelle est peu respectée et ce RFC explique pourquoi et documente la réaction la plus courante des mises en œuvre de TCP.

Soit une machine qui a une connexion TCP (RFC 793 [1]) avec une autre (ou bien qui est en train de tenter d'établir cette connexion). Arrive un paquet ICMP (RFC 792) qui dit, par exemple "*Network unreachable*". Que doit faire TCP ? Abandonner la connexion ? Ignorer le message ?

Les exigences pour les machines Internet, le RFC 1122, sont claires. Il existe deux sortes d'erreur ICMP, les « douces » ("*soft errors*", qui sont en général temporaires, c'est le cas du type 3, "*Destination unreachable*", pour ses codes 0, "*Network unreachable*", 1, "*Host unreachable*" et 5 "*Source route failed*") et les « dures » ("*hard errors*", en général de longue durée, comme le type 3 pour le code 3, "*Port unreachable*", dû en général à l'action délibérée d'un pare-feu). Le RFC 1122 dit que TCP ne doit **pas** interrompre une connexion pour une erreur douce. En effet, celles-ci étant souvent transitoires, une telle interruption immédiate empêcherait de tirer profit de la résilience de l'Internet, où le réseau se reconfigure pour contourner un problème (voir aussi la section 2.2).

Pourtant, la plupart des mises en œuvre de TCP ignorent cette règle, afin d'éviter une longue attente au cas où une machine ne soit pas joignable sur certaines adresses (cas courant, notamment avec IPv6). Notre RFC 5461, sans changer la norme officielle, documente le comportement non-officiel, explique ses raisons et indique ses limites.

D'abord, le cas couvert par ce nouveau RFC est uniquement celui de l'**établissement** de la connexion (quand TCP est dans l'état `SYN_RECEIVED` ou `SYN_SENT`). Celui des connexions déjà établies n'est pas traité.

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc793.txt`

Le RFC 816 séparait le traitement des pannes réseaux en deux : leur **détection** et leur **réparation**. ICMP est un outil de détection. Il permet de voir qu'il y a un problème. Mais quelle réparation utiliser ? (section 1 du RFC).

Le RFC 1122, section 4.3.2.9, dit que TCP ne doit pas couper la connexion pour les erreurs douces (et, par contre, y mettre fin en cas d'erreur dure). La section 2 de notre RFC revient sur cette règle en notant qu'elle n'est absolument pas respectée en pratique.

Pourquoi ? C'est l'objet de la section 3. En gros, si on suit le RFC 1122 au pied de la lettre, et qu'on essaie de se connecter à une machine qui a plusieurs adresses IP (ce qui est courant avec IPv6), l'algorithme typique de l'application est séquentiel : on essaie toutes les adresses l'une après l'autre. Si la réception d'une erreur ICMP n'entraine pas l'échec immédiat d'une tentative sur une adresse, il faudra attendre l'expiration du délai de garde (trois minutes, dit le RFC 1122 !) pour renoncer à cette adresse et essayer la suivante (section 3.1). Le cas fréquent d'une adresse IPv4 et d'une IPv6 sur la même machine est discuté en section 3.2.

Alors, que font les mises en œuvre actuelles de TCP ? La section 4 décrit deux stratégies courantes. Le RFC se sent obligé de rappeler que ces statégies sont, formellement, une violation de la norme mais changer celle-ci semble difficile, alors autant documenter les écarts...

La première stratégie (section 4.1), est de réagir différemment selon que l'erreur est reçue pendant la phase d'établissement de la connexion ou bien après. Le noyau Linux, depuis la version 2.0.0, renonce immédiatement à l'établissement d'une connexion TCP lorsqu'il reçoit le paquet ICMP pendant ledit établissement. L'application peut alors immédiatement passer à l'adresse suivante.

La seconde stratégie (section 4.2) est plus méfiante. Elle consiste à ne renoncer à l'établissement de la connexion qu'après N erreurs ICMP, avec N ¿ 1 (il vaut exactement un dans la stratégie précédente). C'est la méthode utilisée par FreeBSD. (Par contre, je ne trouve pas comment régler N sur FreeBSD 7.0.)

Évidemment, violer la norme n'est pas innocent. La section 5 détaille donc les conséquences qui attendent le violeur. Par exemple, 5.1 explique qu'un problème temporaire qui affecte **toutes** les adresses de destination risque d'empêcher complètement la création d'une nouvelle connexion, alors qu'un TCP conforme à la norme aurait patienté et serait finalement passé. Il y a ici clairement un compromis entre l'attente, dans l'espoir de finalement réussir (parfait pour les connexions non-interactives comme un transfert de fichier nocturne avec rsync lancé par cron) et l'abandon rapide (sans doute préférable pour les applications interactives, où il ne faut pas bloquer l'utilisateur humain).