RFC 6928 : Increasing TCP's Initial Window

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 30 avril 2013

Date de publication du RFC : Avril 2013

http://www.bortzmeyer.org/6928.html

Sur l'Internet, il y a des choses qu'on hésite à toucher. Depuis quelques épisodes fameux de « catastrophes congestives », où tout l'Internet s'est arrêté en raison de l'encombrement des tuyaux, la mémoire collective du réseau, incarnée notamment par l'IETF, voit avec une extrême méfiance toute modification des algorithmes de TCP, qui sont la principale ligne de défense de l'Internet face à la congestion. Néanmoins, le réseau évolue, les choses changent et il y a des gens qui insistent pour essayer des changements. Cela fait un certain temps que Google réclame <http://googlecode.blogspot.fr/2012/01/lets-make-tcp-faster.html> une modification connue sous le doux nom de IW10 (pour "*Initial Window 10 segments*") et ce RFC est le premier à la documenter officiellement, après trois ans de discussion dans le groupe de travail. Pour l'instant, c'est encore étiqueté comme « expérimental ».

La norme reste celle du RFC 3390[1]. Elle dit que la fenêtre initiale d'une connexion TCP, à savoir le nombre maximal d'octets qu'on peut envoyer avant le premier accusé de réception, est donnée par une formule qui, en pratique est d'environ 4 kilo-octets, soit deux à trois segments (paquets TCP), parfois quatre. Voici, vu par tcpdump sur le client, un exemple entre deux machines séparées par 100 ms de RTT, un client HTTP qui demande un gros fichier et un serveur :

```
21:54:15.371487 IP 204.62.14.153.80 > 192.168.2.1.57886: Flags [.], seq 1:1449, ack 118, win 1810,
options [nop,nop,TS val 1079804426 ecr 2768078], length 1448
21:54:15.371509 IP 192.168.2.1.57886 > 204.62.14.153.80: Flags [.], ack 1449, win 137, options [nop
,nop,TS val 2768109 ecr 1079804426], length 0
21:54:15.372574 IP 204.62.14.153.80 > 192.168.2.1.57886: Flags [.], seq 1449:2897, ack 118, win 181
0, options [nop,nop,TS val 1079804426 ecr 2768078], length 1448
21:54:15.372593 IP 192.168.2.1.57886 > 204.62.14.153.80: Flags [.], ack 2897, win 182, options [nop
,nop,TS val 2768109 ecr 1079804426], length 0
21:54:15.372615 IP 204.62.14.153.80 > 192.168.2.1.57886: Flags [.], seq 2897:4345, ack 118, win 181
0, options [nop,nop,TS val 1079804426 ecr 2768078], length 1448
```

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc3390.txt

```
21:54:15.372630 IP 192.168.2.1.57886 > 204.62.14.153.80: Flags [.], ack 4345, win 227, options [nop
,nop,TS val 2768109 ecr 1079804426], length 0
21:54:15.372897 IP 204.62.14.153.80 > 192.168.2.1.57886: Flags [P.], seq 4345:5793, ack 118, win 18
10, options [nop,nop,TS val 1079804426 ecr 2768078], length 1448
21:54:15.372910 IP 192.168.2.1.57886 > 204.62.14.153.80: Flags [.], ack 5793, win 273, options [nop
,nop,TS val 2768109 ecr 1079804426], length 0
21:54:15.488019 IP 204.62.14.153.80 > 192.168.2.1.57886: Flags [.], seq 5793:7241, ack 118, win 181
0, options [nop,nop,TS val 1079804455 ecr 2768109], length 1448
```

On voit que `204.62.14.153` a envoyé quatre segments avant de devoir s'arrêter, les accusés de réception n'ayant pas eu le temps de l'atteindre. La pause dure 116 ms, avant que le serveur ne reprenne son envoi.

Ce nombre de segments est trop petit (bien plus petit que les tampons d'entrée/sortie d'un équipement réseau typique) car, sur les réseaux modernes, cela veut dire qu'un pair TCP qui veut envoyer un gros fichier va très vite remplir cette fenêtre et va devoir attendre ensuite, alors qu'il ne sait absolument pas si le réseau est congestionné ou pas. D'où la proposition IW10 qui est de permettre l'envoi de dix segments. (Pourquoi dix ? Comme l'explique l'annexe A, c'est le résultat de tests avec plusieurs valeurs possibles. Dix était la valeur qui était le meilleur compromis entre les gains et les inconvénients. Regardez le dessin 2 dans l'article "*An Argument for Increasing TCP's Initial Congestion Window*" si vous avez besoin d'être convaincu : il montre un progrès jusqu'à 16 et une baisse ensuite. Notez qu'une des techniques étudiées par le groupe de travail - et finalement abandonnée - avait été un système complexe de mesure et de rétroaction, avec des tailles initiales de fenêtre variables.)

La formule proposée par ce RFC (section 2) est fenêtre_initiale (en octets) = minimum (10*MSS, maximum (2*MSS, 14600)). Cela pourra faire jusqu'à dix segments, en fonction de la MSS. Naturellement, la machine aura le droit de choisir une fenêtre plus petite, ce choix n'est en effet pas dangereux ; mais pas de fenêtre plus grande, le but étant d'éviter la congestion de réseau.

Notez bien que cette formule ne s'applique qu'au début de la connexion. Lorsque TCP reprend son algorithme de démarrage en douceur ("*slow start*"), par exemple suite à des pertes de paquets, il doit s'en tenir au RFC 2581.

Il y a aussi quelques questions pratiques à se poser. Par exemple, une connexion TCP est bi-directionnelle : ce n'est pas tout d'utiliser une fenêtre plus grande à l'émission, il faut aussi annoncer à son pair une fenêtre de réception plus grande (cf. « "*An Argument for Increasing TCP's Initial Congestion Window*" <http://research.google.com/pubs/pub36640.html> » de Dukkipati, N., Refice, T., Cheng, Y., Chu, J., Sutin, N., Agarwal, A., Herbert, T. et J. Arvind, article de 2010 qui va être souvent cité ici et dont je recommande fortement la lecture. À l'époque de l'article, la plupart des systèmes annonçaient cette grande fenêtre, sauf Linux.)

Ce changement a des conséquences pour les applications qui utilisent TCP (section 3). Ainsi, HTTP 1.1 n'autorisait que deux connexions TCP simultanées vers le même serveur (RFC 2616, section 8.1.4, mais le RFC 7230 a libéralisé cette consigne). Les navigateurs en ouvrent en général plus que cela, pour éviter l'attente due à la faible taille de la fenêtre TCP. (Voir « "*A Swifter Start for TCP*" <http://www.icir.org/mallman/papers/tr8339.ps> », par Partridge, C., Rockwell, D., Allman, M., Krishnan, R. et J. Sterbenz, rapport technique n° 8339 de BBN en 2002.) C'est égoïste de leur part (cela s'oppose aux mécanismes de contrôle de congestion de TCP) et cela ne devrait logiquement plus être nécessaire avec le déploiement d'IW10. Le RFC conseille donc de réduire le nombre de connexions HTTP simultanées.

La section 4 du RFC donne des éléments de contexte et d'histoire. Au début (cf. le célébrissime article de Van Jacobson, « "*Congestion Avoidance and Control*" <http://ee.lbl.gov/papers/congavoid.

pdf> », en 1998), TCP ne pouvait envoyer qu'un seul segment avant de recevoir le premier accusé de réception. Ce n'était pas un problème pour de longues communications, comme celles de telnet. Mais l'Internet aujourd'hui voit surtout du trafic Web (cf. « "*Atlas Internet Observatory 2009 Annual Report*" <http://www.nanog.org/meetings/nanog47/presentations/Monday/Labovitz_ObserveReport_N47_Mon.pdf> » de Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J. Jahanian, F. et M. Karir, présenté à NANOG 47 en 2009.), avec des connexions de courte durée. Attendre longtemps avant de pouvoir envoyer plusieurs segments est très pénalisant (la plupart de ces connexions courtes ne quitteront jamais l'état TCP « démarrage en douceur »).

Surtout, les changements dans la capacité des réseaux ont été spectaculaires. Le rapport « "*The State of the Internet*" <http://www.akamai.com/html/about/press/releases/2010/press_011310_1.html> » d'Akamai en 2010, annonce que les accès à plus de 2 Mb/s forment désormais la majorité, ce qui met la capacité moyenne à 1,7 Mb/s, alors que, pendant ce temps, la fenêtre d'envoi initiale de TCP restait aux 4 ko du RFC 2414, sans bouger pendant dix ans. Avec un RTT de 200 ms, cette fenêtre ne permettait que 200 kb/s.

Des tas de changements avaient été proposés (pour ne citer que les RFC, voir RFC 4782 et RFC 6077) mais sans être réellement déployés. Ce sont donc les applications qui se sont adaptées, comme vu plus haut avec les navigateurs qui ouvrent plusieurs connexions, et les sites Web créant des sous-domaines sans autre raison que de débrayer la limitation du nombre de connexions par serveur (cf. « "*A Client-Side Argument For Changing TCP Slow Start*" <http://sites.google.com/a/chromium.org/dev/spdy/An_Argument_For_Changing_TCP_Slow_Start.pdf> » en 2010). Le déploiement d'IW10 devrait logiquement s'accompagner d'un changement de stratégie de ces navigateurs Web.

Au fait, en parlant d'HTTP, les connexions persistentes de HTTP 1.1 ne devraient-elles pas résoudre le problème ? Le problème est que les mesures faites avec Chrome (article de Dukkipati et al. cité plus haut) montrent que 35 % des requêtes sont faites sur des nouvelles connexions, ne pouvant pas tirer profit des connexions persistentes.

Bon, mais, au fait, quels sont les avantages exacts qu'il y a à agrandir la fenêtre initiale ? La section 5 les examine en détail. D'abord, on réduit la latence. Par exemple, si on a 21 segments à envoyer, IW3 (fenêtre initiale de trois segments) va nécessiter 4 aller-retour (envoi de données + attente de l'accusé de réception) contre 2 pour IW10. Ensuite, une telle augmentation est cohérente avec l'augmentation régulière de taille des objets. Par exemple, sur le Web, les études citées par le RFC (mesures faites sur le moteur de recherche Google) montrent que seules 10 % des ressources sur le Web tiennent encore dans les 4 ko du RFC 3390. En outre, si on utilise les connexions persistentes de HTTP 1.1, on aura encore plus de données à transmettre sur une connexion HTTP. Le Web grossit, il est logique que la fenêtre initiale de TCP suive le mouvement.

Et il n'y a pas d'inconvénients à augmenter la taille de la fenêtre initiale ? Les sections 6 et 7 décrivent le côté obscur. D'abord, en section 6, les risques pour l'utilisateur. Il y a notamment le risque de remplir complètement les tampons d'entrée-sortie dès le démarrage, pour des petits routeurs. Le RFC estime toutefois que ce risque est faible avec les tampons actuels, en s'appuyant sur l'étude « "*Characterizing Residential Broadband Networks*" <http://research.microsoft.com/en-us/um/people/ssaroiu/publications/imc/2007/imc2007-dischinger.pdf> » de Dischinger, M., Gummadi, K., Haeberlen, A. et S. Saroiu, en 2007, qui montre que le routeur typique de M. Michu a au moins 130 ms de tampon.

Ça, c'était pour l'utilisateur. Et pour le réseau (section 7) ? Risque t-on davantage de congestion ? Probablement pas puisque IW10 ne s'applique qu'au début de la connexion. Tout le reste du mécanisme de contrôle de la congestion de TCP reste en place, inchangé, notamment sa réaction aux pertes de paquets. En revanche, il peut y avoir un problème de justice. Tant qu'une partie des machines applique IW10

et qu'une autre partie en est resté au RFC 3390, les machines IW10 vont prendre une part plus grande des ressources réseau (c'est une question générale de l'Internet : la répartition juste entre différentes connexions concurrentes dépend de leur bonne volonté). C'est donc un point à surveiller. (Un autre problème de justice est traité dans l'annexe A, entre des machines toutes IW10 mais où certaines n'ont pas dix segments à envoyer. Les mesures semblent indiquer que ces dernières machines ne sont pas défavorisées.)

Autre inconvénient possible : comme IW10 remplit plus vite les tampons d'entrée/sortie des routeurs, si ceux-ci sont trop grands (phénomène du "*bufferbloat*"), IW10 va aggraver le problème. Les protocoles sensibles à la latence comme le DNS ou certains jeux vont alors souffrir. Les mécanismes proposés comme alternative au "*bufferbloat*" (ECN, AQM, etc) sont alors encore plus nécessaires.

En synthèse (section 8), le RFC estime que le plus gros risque est pour les utilisateurs ayant des liaisons lentes avec des tampons d'entrée/sortie réduits, dans leur "*box*". Les 10 segments initiaux peuvent, dans certains cas, faire déborder ces tampons, le réseau ne pouvant pas les envoyer assez vite. Le RFC suggère, si on détecte qu'on est sur un tel lien, de ne pas envoyer forcément les dix segments autorisés, et d'annoncer une fenêtre de réception plus petite. Voir aussi le RFC 3150, sur le problème général de ces liens lents.

La théorie, c'est bien joli mais, quand elle dit qu'il ne faut pas trop s'inquiéter, il est prudent de vérifier par des mesures effectives, pas juste des raisonnements. C'est ce que fait la section 10. L'expérience a été faite par Google dans deux de ses centres de données, l'un voyant plutôt des clients ayant la capacité réseau moyenne, l'autre ayant en moyenne des utilisateurs de connexions lentes (20 % à moins de 100 kb/s). Avec IW10, la latence a baissé de 11,7 % dans le centre moyen et 8,7 % dans le centre lent. En regardant plus finement, les auteurs ont constaté que, comme l'indiquait la théorie, les bénéfices les plus importants d'IW10 ont été pour les gens situés loin (forte latence réseau) mais ayant des connexions rapides (ce qu'on nomme le BDP élévé avec BDP = "*Bandwidth-Delay Product*"). Mais, contrairement à la théorie, même les clients à capacité réseau limitée ont vu une amélioration. (Voir l'exposé « "*Increasing TCP initial window*" `<http://www.ietf.org/proceedings/78/slides/iccrg-3.pdf>` » de Dukkipati, D., Cheng, Y., Chu, J. et M. Mathis, présenté à l'IETF 78 en 2010.) Aucun problème grave n'a été observé.

D'autres expériences ont été faites, listées sur le site Web du projet IW10 `<http://code.google.com/speed/protocols/tcpm-IW10.html>` ou en section 11 de ce RFC. Notez toutefois que la plupart sont des simulations, pas des mesures sur le vrai Internet. Mais ce n'est pas forcément très grave, l'étude de Google, par son ampleur, a semblé largement suffisante. Bien des RFC ont été adoptés sans une aussi grande étude préalable !

L'annexe A du RFC contient une liste d'inquiétudes théoriques sur IW10 et les résultats des tests spécifiquement liés à chacune de ces inquiétudes. Ainsi, le taux de pertes de paquets n'a pas augmenté lors des tests, sauf lors de cas extrêmes où, même avec les valeurs du RFC 3390, le phénomène de pertes se produisait. De même, des mesures ont été faites en Afrique, pas seulement en Europe et en Amérique du Nord, pour s'assurer qu'IW10 n'aurait pas d'effet pénible pour les liens moins favorisés.

En conclusion, le RFC recommande (section 12) d'activer IW10 **mais** que les fournisseurs de mises en œuvre de TCP et/ou les administrateurs réseau surveillent le trafic, à la recherche d'indicateurs comme le taux de pertes de paquet, pour détecter une éventuelle augmentation des phénomènes négatifs.

Ce changement de la fenêtre initiale est déjà apparue dans certains systèmes. Par exemple, pour Linux, cela a été fait en 2011 dans le "*commit*" git `442b9635c569fef038d5367a7acd906db4677ae1`. Notez que, sur Linux, `ip route ... initcwnd N` vous permet d'ajuster cette variable, pour expérimenter.

Un peu de lecture supplémentaire ? Le premier exposé sur IW10 date de 2010 `<http://www.ietf.org/proceedings/77/slides/tcpm-4.pdf>`. Une argumentation hostile à IW10 ? Vous pouvez lire l'"*Internet-Draft*" `draft-gettys-iw10-considered-harmful`.