Groupe de travail Réseau
Request for Comments : 3390
RFC mise à jour : 2581
RFC rendue obsolète : 2414
Catégorie : En cours de normamisation

M. Allman, BBN/NASA GRC
S. Floyd, ICIR
C. Partridge, BBN Technologies
octobre 2002
Traduction Claude Brière de L'Isle

Augmentation de la fenêtre initiale de TCP

Résumé
Le présent document spécifie une norme facultative pour que TCP augmente la fenêtre initiale permise d'un ou deux segments à environ 4 000 octets, remplaçant la RFC2414. Il expose les avantages et les inconvénients d'une plus grande fenêtre initiale, et comporte la discussion des expériences et simulations montrant qu'une plus forte fenêtre initiale ne conduit pas à un collapsus d'encombrement. Enfin, ce document donne des lignes directrices sur les questions de mise en œuvre.

Terminologie
Dans le présent document, les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" sont à interpréter comme décrit dans le BCP 14, [RFC2119].

Table des Matières

1. Modification à TCP

Le présent document rend obsolète la [RFC2414] et met à jour la [RFC2581] et spécifie une augmentation de la limite supérieure permise pour la fenêtre initiale de TCP de un ou deux segments à deux à quatre segments. Dans la plupart des cas, ce changement résulte en une limite supérieure de la fenêtre initiale d'environ 4 koctets (bien que dans le cas d'une grande taille de segment, la fenêtre initiale permise de deux segments puisse être significativement supérieure à 4 koctets).

La limite supérieure pour la fenêtre initiale est données plus précisément en (1):

min (4*MSS, max (2*MSS, 4380 octets)) (1)

Note : L'envoi d'un paquet de 1500 octets indique une taille maximum de segment (MSS, *Maximum Segment Size*) de 1460 octets (en supposant qu'il n'y a pas d'option IP ou TCP). Donc, limiter la MSS de la fenêtre initiale à 4 380 octets permet à l'envoyeur de transmettre trois segments initialement dans le cas courant lorsque il utilise des paquets de 1500 octets.

De façon équivalente, la limite supérieure de la taille de fenêtre initiale se fonde sur la MSS, comme suit :

Si (MSS ≤ 1095 octets)
alors fen ≤ 4 * MSS ;
Si (1095 octets < MSS < 2190 octets)
alors fen ≤ 4380;
Si (2190 octets ≤ MSS)
alors fen ≤ 2 * MSS ;

Cette fenêtre initiale augmentée est facultative : une mise en œuvre de TCP PEUT commencer avec une fenêtre initiale plus grande. Cependant, on s'attend à ce que les mises en œuvre les plus courantes de TCP choisissent d'utiliser une plus grande fenêtre initiale d'encombrement étant donnée l'équation (1) ci-dessus.

Cette limite supérieure de taille de fenêtre initiale représente un changement par rapport à la [RFC2581], qui spécifiait que la fenêtre d'encombrement soit initialisée à un ou deux segments.

Ce changement s'applique à la fenêtre initiale de la connexion dans le premier délai d'aller-retour (RTT, *round trip time*) de la transmission de données qui suit la prise de contact en trois étapes de TCP. Ni le SYN/ACK ni son accusé de réception (ACK) dans la prise de contact en trois étapes ne devrait augmenter la taille de la fenêtre initiale au delà de ce qui est présenté dans l'équation (1). Si le SYN ou SYN/ACK est perdu, la fenêtre initiale utilisée par un envoyeur après un SYN correctement transmis DOIT être d'un segment consistant en MSS octets.

Les mises en œuvre de TCP utilisent le démarrage lent de trois façons différentes : (1) pour démarrer une nouvelle connexion (la fenêtre initiale) ; (2) pour redémarrer la transmission après une longue période d'inactivité (la fenêtre de redémarrage) ; et (3) pour redémarrer la transmission après une fin de temporisation de retransmission (la fenêtre de perte). Le changement spécifié dans le présent document affecte la valeur de la fenêtre initiale. Facultativement, une mise en œuvre de TCP PEUT régler la fenêtre de redémarrage au minimum de la valeur utilisée pour la fenêtre initiale et de la valeur actuelle de cwnd (en d'autres termes, utiliser une plus grande valeur pour la fenêtre de redémarrage ne devrait jamais augmenter la taille de cwnd). Ces changements NE CHANGENT PAS la fenêtre de perte, qui doit rester d'un segment de MSS octets (pour permettre la plus petite taille de fenêtre possible dans le cas d'encombrement sévère).

2. Questions de mise en œuvre

Lorsque de plus grandes fenêtres initiales sont mises en œuvre avec la découverte de la MTU de chemin [RFC1191], et que la MSS utilisée se trouve trop grande, la fenêtre d'encombrement 'cwnd' DEVRAIT être réduite pour empêcher de grosses salves de plus petits segments. Précisément, 'cwnd' DEVRAIT être réduit du ratio de l'ancienne taille de segment sur la nouvelle taille de segment.

Lorsque de plus grandes fenêtres initiales sont mises en œuvre avec la découverte de la MTU de chemin [RFC1191], les solutions de remplacement sont de régler à 1 le bit "Ne pas fragmenter" (DF) dans tous les segments de la fenêtre initiale, ou d'établir le bit "Ne pas fragmenter" (DF) dans un des segments. La question de savoir laquelle de ces deux solutions est la meilleure reste ouverte ; on espère que les expériences de mise en œuvre pourront éclaircir cette question. Dans le premier cas d'établir le bit DF dans tous les segments, si les paquets initiaux sont trop grands, alors tous les paquets initiaux seront éliminés dans le réseau. Dans le second cas qui est d'établir le bit DF dans un seul segment, si les paquets initiaux sont trop grands, alors tous les paquets initiaux sauf un seront fragmentés dans le réseau. Lorsque le second cas est suivi, établir le bit DF dans le dernier segment dans la fenêtre initiale donne une moindre chance de retransmissions inutiles lorsque la taille du segment initial se trouve être trop grande, parce qu'elle minimise les chances d'ACK dupliqués qui déclanchent une retransmission rapide. Cependant, il faut prêter attention à l'interaction entre de plus grandes fenêtres initiales et la découverte de la MTU de chemin.

La plus grande fenêtre initiale spécifiée dans le présent document n'est pas destinée à encourager les navigateurs de la Toile à ouvrir plusieurs connexions TCP simultanées, toutes avec de grandes fenêtres initiales. Lorsque des navigateurs de la Toile ouvrent des connexions TCP simultanées pour la même destination, elles travaillent contre le mécanisme de contrôle d'encombrement de TCP [FF99], sans considération de la taille de la fenêtre initiale. Combiner ce comportement avec de plus grandes fenêtres initiales augmente encore l'injustice à l'égard des autres trafic du réseau. On suggère l'utilisation de

HTTP/1.1 [RFC2068] (connexions TCP persistantes et travail en parallèle) comme moyen de réaliser de meilleures performances des transferts sur la Toile.

3. Avantages d'une plus grande fenêtre initiale

1. Lorsque la fenêtre initiale est d'un segment, un receveur qui emploie des ACK retardés [RFC1122] est forcé d'attendre une fin de temporisation pour générer un ACK. Lors d'une fenêtre initiale d'au moins deux segments, le receveur va générer un ACK après qu'arrive le second segment de données. Cela élimine l'attente de la fin de temporisation (souvent jusqu'à 200 ms, et éventuellement jusqu'à 500 ms [RFC1122]).

2. Pour les connexions qui transmettent seulement une petite quantité de données, une plus grande fenêtre initiale réduit le temps de transmission (en supposant des taux d'abandon de segment plus modérés). Pour de nombreux transferts de messagerie électronique (SMTP [RFC0821]) et de page de la Toile (HTTP [RFC1945], [RFC2068]) qui font moins de 4 koctets, la plus grande fenêtre initiale va réduire le temps de transfert des données à un seul RTT.

3. Pour les connexions qui sont capables d'utiliser de grandes fenêtre d'encombrement, cette modification élimine jusqu'à trois RTT et une temporisation d'ACK retardé durant la phase initiale de démarrage lent. Cela présente un avantage particulier pour les connexions TCP à forte bande passante et fort délai de propagation, telles que celles qui sont sur des liaisons par satellite.

4. Inconvénients d'une plus grande fenêtre initiale pour les connexions individuelles

Dans les environnements de fort encombrement, en particulier pour les routeurs qui ont un biais contre le trafic sporadique (comme dans les files d'attentes de routeur à abandon de la queue) une connexion TCP peut parfois avoir intérêt à commencer avec une fenêtre initiale d'un seul segment. Il y a des scénarios où une connexion TCP qui fait un démarrage lent à partir d'une fenêtre initiale de un segment peut n'avoir pas de segment éliminé, alors qu'une connexion TCP qui démarre avec une fenêtre initiale de quatre segments peut rencontrer des retransmissions inutile à cause de l'incapacité du routeur à traiter les petites salves. Il peut en résulter des fins de temporisations de retransmission inutiles. Pour une connexion à grande fenêtre qui est capable de récupérer sans temporisation de retransmission, il peut en résulter une transition précoce inutile du démarrage lent à la phase d'évitement d'encombrement de l'algorithme d'augmentation de fenêtre. Ces abandons de segments prématurés ne vont probablement pas se produire dans des réseaux non encombrés avec une mémoire tampon suffisante ou dans des réseaux modérément encombrés où le routeur encombré utilise la gestion active de file d'attente (comme la détection précoce aléatoire [FJ93], [RFC2309]).

Certaines connexions TCP vont avoir de meilleures performances avec la plus grande fenêtre initiale même si la sporadicité de la fenêtre initiale résulte en abandons de segment prématurés. Cela sera vrai si (1) la connexion TCP récupère de l'abandon de segment sans temporisation de retransmission, et (2) la connexion TCP est en fin de compte limitée à une petite fenêtre d'encombrement soit par l'encombrement du réseau, soit par la fenêtre annoncée du receveur.

5. Inconvénients d'une plus grande fenêtre initiale pour le réseau

En termes de collapsus d'encombrement potentiel, on considère deux dangers potentiels distincts pour le réseau. Le premier danger serait un scénario où un grand nombre de segments sur des liaisons encombrées seraient dupliqués alors qu'ils ont déjà été reçus par le receveur. Le second danger serait un scénario où un grand nombre de segments sur des liaisons encombrées seraient des segments qui seraient éliminés plus tard dans le réseau avant d'atteindre leur destination finale.

En termes d'effet négatif sur d'autre trafic du réseau, un inconvénient potentiel des grandes fenêtres initiales serait qu'elles augmentent le taux général d'abandon de paquet dans le réseau. On expose ces trois questions ci-dessous.

Segments dupliqués :
Comme décrit au paragraphe précédent, la plus grande fenêtre initiale pourrait occasionnellement résulter en un abandon de segment de la fenêtre initiale, lorsque ce segment pourrait n'avoir pas été éliminé si l'envoyeur avait fait un démarrage lent à partir d'une fenêtre initiale d'un seul segment. Cependant, l'Appendice A montre que même dans ce cas, la plus grande fenêtre initiale n'aurait pas résulté en la transmission d'un grand nombre de segments dupliqués.

Segments éliminés ultérieurement dans le réseau :
De combien la plus grande fenêtre initiale de TCP augmente t-elle le nombre de segments sur des liaisons encombrées qui seraient éliminés avant d'atteindre leur destination finale ? C'est un problème qui ne peut survenir que pour des connexions

qui ont plusieurs liaisons encombrées, où des segments peuvent utiliser une bande passante raréfiée sur la première liaison encombrée le long du chemin, pour être finalement éliminés plus loin sur le chemin.

D'abord, beaucoup de connexions TCP auront seulement une liaison encombrée sur le chemin. Les segments éliminés de ces connexions ne "gâchent" pas une bande passante raréfiée, et ne contribuent pas au collapsus d'encombrement.

Cependant, certains chemins du réseau vont avoir plusieurs liaisons encombrées, et les segments éliminés de la fenêtre initiale pourraient utiliser une bande passante raréfiée le long des premières liaisons encombrées avant d'être finalement éliminés sur des liaisons encombrées ultérieures. Dans la mesure où le taux d'abandon est indépendant de la fenêtre initiale utilisée par les segments TCP, le problème des liaisons encombrées portant des segments qui vont être éliminés avant d'atteindre leur destination sera similaire pour les connexions TCP qui commencent par envoyer quatre segments ou un segment.

Taux accru d'élimination de paquet :
Pour un réseau qui a un fort taux d'abandon de segment, augmenter la fenêtre initiale TCP pourrait augmenter encore le taux d'abandon de segment. Ceci en partie parce que les routeurs qui font la gestion de file d'attente avec abandon de la queue ont des difficultés avec le trafic sporadique en temps d'encombrement. Cependant, comme les arrivées sont non corrélées pour les connexions TCP, la plus grande fenêtre initiale TCP ne devrait pas augmenter significativement le taux d'abandon de segment. Les explorations fondées sur la simulation de ces questions sont exposées au paragraphe 7.2.

Ces dangers potentiels pour le réseau sont examinés dans des simulations et des expériences décrites dans la section suivante. Notre opinion est que bien qu'il y ait des dangers de collapsus d'encombrement dans l'Internet actuel (voir dans [FF99] une discussion des dangers de collapsus d'encombrement résultant d'un déploiement accru de connexions UDP sans contrôle d'encombrement de bout en bout) il n'existe pas un tel danger pour le réseau d'augmenter la fenêtre initiale TCP à 4 koctets.

6. Interactions avec le temporisateur de retransmission

Utiliser une plus grande salve initiale de données peut exacerber les problèmes existants avec des temporisations de retransmission parasites sur des chemins à faible bande passante, en supposant l'algorithme standard pour déterminer la temporisation de retransmission (RTO, *retransmission timeout*) TCP [RFC2988]. Le problème est que sur les réseaux à faible bande passante sur lesquels le temps de transmission d'un paquet est une large portion du délai d'aller-retour, les petits paquets utilisés pour établir une connexion TCP ne génèrent pas une estimation appropriée de RTO. Lorsque la première fenêtre de paquets de données est transmise, le temporisateur de retransmission de l'envoyeur pourrait arriver à expiration avant que soient reçus les accusés de réception de ces paquets. Lorsque chaque accusé de réception arrive, le temporisateur de retransmission est généralement remis à zéro. Donc, le temporisateur de retransmission ne va pas arriver à expiration tant qu'un accusé de réception arrive au moins une fois par seconde, étant donné le minimum d'une seconde recommandé pour le RTO dans la [RFC2988].

Par exemple, considérons une liaison à 9,6 kbit/s. La mesure du RTT initial sera de l'ordre de 67 ms, si on considère simplement le temps de transmission de deux paquets (le SYN et le SYN-ACK) chacun consistant en 40 octets. En utilisant l'estimateur de RTO de la [RFC2988], cela donne un RTO initial de 201 ms (67 + 4*(67/2)). Cependant, on arrondit le RTO à 1 seconde, comme spécifié dans la RFC2988. Supposons ensuite qu'on envoie une fenêtre initiale de un ou plusieurs paquets de 1500 octets (1460 octets de données plus les "frais généraux"). Chaque paquet va prendre de l'ordre de 1,25 secondes pour la transmission. Donc, le RTO va arriver à expiration avant que l'ACK pour le premier paquet revienne, causant une fin de temporisation parasite. Dans ce cas, une plus grande fenêtre initiale de trois ou quatre paquets exacerbe les problèmes causés par cette fin de temporisation parasite.

Une façon de traiter ce problème est de rendre l'algorithme de RTO plus prudent. Durant la fenêtre initiale de données, par exemple, le RTO pourrait être mis à jour pour chaque accusé de réception reçu. De plus, si le temporisateur de retransmission arrive à expiration pour un paquet perdu lors de la première fenêtre de données, on pourrait laisser courir le retard exponentiel du temporisateur de retransmission au moins jusqu'à une mesure valide de RTT, ce qui implique qu'un paquet de données soit reçu.

Une autre méthode serait de s'interdire de prendre un échantillon de RTT durant l'établissement de la connexion, laissant le RTO par défaut en place jusqu'à ce que TCP prenne un échantillon à partir d'un segment de données et de l'ACK correspondant. Bien que cette méthode aide vraisemblablement à empêcher les retransmissions parasites, elle peut aussi ralentir le transfert des données si des pertes surviennent avant que le RTO soit généré. L'utilisation de la transmission limitée de la [RFC3042] pour aider une connexion TCP à récupérer de pertes en utilisant la retransmission rapide plutôt que le temporisateur de RTO atténue la dégradation des performances causée par l'utilisation d'un RTO par défaut élevé durant la fenêtre initiale de la transmission des données.

La présente spécification ne donne pas de conclusion sur la décision de la mise en œuvre (s'il en est une) en ce qui concerne le RTO, lors de l'utilisation d'une plus grande fenêtre initiale. Cependant, l'approche RECOMMANDÉE est de s'interdire d'échantillonner le RTT durant la prise de contact en trois étapes, et de garder le RTO par défaut en place jusqu'à ce que soit pris un échantillon de RTT impliquant un paquet de données. De plus, il est RECOMMANDÉ que les mises en œuvre de TCP utilisent la transmission limitée de la [RFC3042].

7. Niveaux typiques de sporadicité pour le trafic TCP

Les plus grandes fenêtres initiales TCP n'augmenteraient pas de façon dramatique la sporadicité du trafic TCP dans l'Internet d'aujourd'hui, parce qu'un tel trafic est déjà très sporadique. Des salves de deux et trois segments sont déjà normales dans TCP [Flo97] ; un ACK retardé (couvrant deux segments non acquittés précédemment) reçu durant l'évitement d'encombrement cause le glissement de la fenêtre d'encombrement et l'envoi de deux segments. Le même ACK retardé reçu durant le démarrage lent cause le glissement de la fenêtre de deux segments et ensuite elle est incrémentée de un segment, résultant en une salve de trois segments. Bien que ce ne soit pas nécessairement normal, des salves de quatre et cinq segments pour TCP ne sont pas rares. En supposant des ACK retardés, un seul ACK éliminé est cause que l'ACK suivant va couvrir quatre segments non acquittés précédemment. Durant l'évitement d'encombrement, cela conduit à une salve de quatre segments, et durant un démarrage lent, une salve de cinq segments est générée.

Il y a aussi des changements en préparation qui réduisent les problèmes de performances posés pas les salves de trafic modérées. Un de ces changements est le déploiement de liaisons à plus grande vitesse dans certaines parties du réseau, où une salve de 4 koctets peut représenter une petite quantité de données. Un second changement, pour les routeurs qui ont suffisamment de mémoire tampon, est le déploiement de mécanismes de gestion de file d'attente tels que la détection précoce aléatoire (RED, *Random Early Detection*) qui est conçue pour être tolérante aux salves de trafic temporaires.

8. Résultats de simulations et d'expériences

8.1 Études des connexions TCP en utilisant cette plus grande fenêtre initiale

La présente section passe en revue des simulations et expériences qui explorent les effets de plus grandes fenêtres initiales sur les connexions TCP. Le premier ensemble d'expériences explore les performances sur les liaisons par satellite. Les plus grandes fenêtres initiales se sont révélées améliorer les performances des connexions TCP sur les canaux par satellite [All97b]. Dans cette étude, une fenêtre initiale de quatre segments (MSS de 512 octets) résultait en une amélioration du débit allant jusqu'à 30 % (selon la taille du transfert). [KAGT98] montre que l'utilisation de plus grandes fenêtres initiales résulte en une diminution du temps de transfert dans des essais de HTTP sur le système par satellite ACTS. Une étude impliquant des simulations d'un grand nombre de transactions HTTP sur un hybride fibre coaxial (HFC) indique que l'utilisation de plus grandes fenêtres initiales diminue le temps nécessaire pour charger des pages de la Toile mondiale [Nic98].

Un second ensemble d'expériences a exploré les performances de TCP sur des liaisons modem à numérotation. Dans des expériences sur un canal à numérotation à 28,8 bit/s [All97a], [AHO98], une fenêtre initiale de quatre segments diminuait le temps de transfert d'un fichier de 16 ko d'environ 10 %, sans être accompagné d'une augmentation du taux d'abandons. Une simulation [RFC2416] étudiait les effets de l'utilisation d'une plus grande fenêtre initiale sur un hôte connecté par une liaison modem lente et un routeur avec une mémoire tampon de 3 paquets. L'étude concluait que pour le scénario retenu, l'utilisation de plus grandes fenêtres initiales n'avait pas d'effet défavorable sur les performances de TCP.

Finalement, [All00] illustre que le pourcentage de connexions à un certain serveur de la Toile qui subissent des pertes dans la fenêtre initiale de transmission de données augmente avec la taille de la fenêtre initiale d'encombrement. Cependant, l'augmentation est en accord avec ce qu'on pourrait attendre de l'envoi d'une plus grosse salve dans le réseau.

8.2 Études des réseaux en utilisant une plus grande fenêtre initiale

Ce paragraphe passe en revue les simulations et expériences qui étudient l'impact de la plus grande fenêtre sur les autres connexions TCP qui partagent le chemin. Les expériences de [All97a], [AHO98] montrent que pour des transferts de 16 ko à 100 hôtes Internet, les fenêtres initiales de quatre segments résultent en une petite augmentation du taux d'abandon de 0,04 segments/transfert. Alors que le taux d'abandon augmente légèrement, le temps de transfert est réduit d'en gros 25 % pour les transferts qui utilisent la fenêtre initiale de quatre segments (MSS de 512 octets) par rapport à une fenêtre initiale de un segment.

Une étude par simulation dans la [RFC2415] explore l'impact d'une grande fenêtre initiale sur du trafic réseau en compétition. Dans cette enquête, des flux HTTP et FTP partagent une seule passerelle encombrée (où le nombre de flux HTTP et FTP varie d'un ensemble de la simulation à l'autre). Pour chaque ensemble de simulations, l'article examine l'utilisation agrégée de la liaison et les taux d'abandon de paquet, le délai médian de page de la Toile, et la puissance du réseau pour les transferts FTP. La plus grande fenêtre initiale résultait généralement en un débit augmenté, des taux légèrement plus élevés d'abandon de paquet, et une augmentation globale de la puissance du réseau. À l'exception d'un scénario, la plus grande fenêtre initiale résultait en une augmentation du taux d'abandon de moins de 1 % de plus que le taux de perte subi lorsque on utilise une fenêtre initiale d'un segment ; dans ce scénario, le taux d'abandon augmentait de 3,5 % avec une fenêtre initiale de un segment, à 4,5 % avec une fenêtre initiale de quatre segments. Les conclusions globales sont qu'augmenter la fenêtre initiale TCP à trois paquets (ou 4380 octets) aide à améliorer les performances perçues.

Morris [Mor97] a étudié les plus grandes fenêtres initiales dans un réseau très encombré avec des transferts d'une taille de 20 ko. Le taux de pertes dans les réseaux où toutes les connexions TCP utilisent une fenêtre initiale de quatre segments se révèle être de 1 à 2 % supérieur à celui d'un réseau où toutes les connexions utilisent une fenêtre initiale de un segment. Cette relation tient dans les scénarios où les taux de pertes avec des fenêtres initiales de un segment vont de 1 % à 11 %. De plus, dans les réseaux où les connexions utilisaient une fenêtre initiale de quatre segments, les connexions TCP passaient plus de temps à attendre l'arrivée à expiration du temporisateur de retransmission (RTO) pour renvoyer un segment qu'elles n'en passaient en utilisant une fenêtre initiale de un segment. Le temps passé à attendre l'arrivée à expiration du temporisateur RTO représente un temps d'inactivité pendant lequel aucun travail utile n'est accompli pour cette connexion. Ces résultats montrent que dans un environnement très encombré, où la part de la bande passante embouteillée de chaque connexion est proche de un segment, utiliser une plus grande fenêtre initiale peut causer une augmentation perceptible en taux de pertes et en temporisations de retransmission.

9. Considérations pour la sécurité

Le présent document discute de la fenêtre d'encombrement initiale permise pour les connexions TCP. Changer cette valeur ne soulève pas de problème de sécurité connu pour TCP.

10. Conclusions

Le présent document spécifie un petit changement à TCP qui va vraisemblablement être bénéfique aux connexions TCP de courte durée et à celles qui sont sur des liaisons avec de longs délais d'aller-retour (économisant plusieurs allers-retours durant la phase initiale de démarrage lent).

11. Remerciements

Nous tenons à remercier Vern Paxson, Tim Shepard, membres de la liste de diffusion Intérêt de bout en bout, et les membres du groupe de travail Mise en œuvre de TCP de l'IETF pour les discussions soutenues de ces problèmes et pour leurs réactions sur le présent document.

12. Références

[AHO98] Mark Allman, Chris Hayes, and Shawn Ostermann, "An Evaluation of TCP with Larger Initial Windows", mars 1998. ACM Computer Communication Review, 28(3), juillet 1998. URL "http://roland.lerc.nasa.gov/~mallman/papers/initwin.ps".

[All97a] Mark Allman. "An Evaluation of TCP with Larger Initial Windows". 40th IETF Meeting -- TCP Implementations WG. décembre 1997. Washington, DC.

[All97b] Mark Allman. "Improving TCP Performance Over Satellite Channels". Master's thesis, Ohio University, juin 1997.

[All00] Mark Allman. "A Web Server's View of the Transport Layer". ACM Computer Communication Review, 30(5), octobre 2000.

[FF96] Fall, K., and Floyd, S., Simulation-based Comparisons of Tahoe, Reno, and SACK TCP. Computer Communication Review, 26(3), juillet 1996.

[FF99] Sally Floyd, Kevin Fall. Promoting the Use of End-to-End Congestion Control in the Internet. IEEE/ACM Transactions on Networking, août 1999. URL "http://www.icir.org/floyd/end2end-paper.html".

[FJ93] Floyd, S., and Jacobson, V., "Random Early Detection gateways for Congestion Avoidance". IEEE/ACM Transactions on Networking, V.1 N.4, août 1993, p. 397-413.

[Flo94] Floyd, S., "TCP and Explicit Congestion Notification". Computer Communication Review, 24(5):10-23, octobre 1994.

[Flo96] Floyd, S., "Issues of TCP with SACK". Technical report, janvier 1996. Disponible à http://www-nrg.ee.lbl.gov/floyd/.

[Flo97] Floyd, S., "Increasing TCP's Initial Window". Viewgraphs, 40th IETF Meeting - TCP Implementations WG. décembre 1997. URL "ftp://ftp.ee.lbl.gov/talks/sf-tcp-ietf97.ps".

[KAGT98] Hans Kruse, Mark Allman, Jim Griner, Diepchi Tran. "HTTP Page Transfer Rates Over Geo-Stationary Satellite Links". mars 1998. Proceedings of the Sixth International Conference on Telecommunication Systems. URL "http://roland.lerc.nasa.gov/~mallman/papers/nash98.ps".

[Mor97] Robert Morris. Communication privée, 1997. Cité uniquement pour remerciements.

[Nic98] Kathleen Nichols. "Improving Network Simulation With Feedback", Proceedings of LCN 98, octobre 1998. URL "http://www.computer.org/proceedings/lcn/8810/8810toc.htm".

[RFC0821] J. Postel, "Protocole simple de transfert de messagerie", STD 10, août 1982.

[RFC1122] R. Braden, "Exigences pour les hôtes Internet – couches de communication", STD 3, octobre 1989. *(MàJ par la* RFC6633)

[RFC1191] J. Mogul et S. Deering, "Découverte de la MTU de chemin", novembre 1990.

[RFC1945] T. Berners-Lee, R. Fielding, H. Frystyk, "Protocole de transfert Hypertext -- HTTP/1.0", mai 1996. *(Information)*

[RFC2068] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, "Protocole de transfert Hypertext -- HTTP/1.1", janvier 1997. (*Obsolète, voir* RFC2616) *(P.S.)*

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.

[RFC2309] B. Braden et autres, "Recommandations sur la gestion de file d'attente et l'évitement d'encombrement dans l'Internet", avril 1998.

[RFC2414] M. Allman, S. Floyd, C. Partridge, "Accroissement de la fenêtre initiale de TCP", septembre 1998. (*Obsolète, voir* RFC3390) *(Expérimentale)*

[RFC2415] K. Poduri, K. Nichols, "Études de simulation d'accroissement de taille initiale de fenêtre TCP", septembre 1998. *(Information)*

[RFC2416] T. Shepard, C. Partridge, "Lorsque TCP commence par quatre paquets dans seulement trois mémoires tampon", septembre 1998. *(Information)*

[RFC2581] M. Alman, V. Paxson et W. Stevens, "Contrôle d'encombrement avec TCP", avril 1999. *(Obsolète, voir* RFC5681)

[RFC2821] J. Klensin, éditeur, "Protocole simple de transfert de messagerie", STD 10, avril 2001. *(Obsolète, voir* RFC5321*)*

[RFC2988] V. Paxson, M. Allman, "Calcul du temporisateur de retransmission de TCP", novembre 2000. *(P.S.)(Obs., voir* RFC6298)

[RFC3042] M. Allman, H. Balakrishnan, S. Floyd, "Amélioration de la récupération de perte dans TCP avec la transmission limitée", janvier 2001. *(P.S.)*

[RFC3168] K. Ramakrishnan et autres, "Ajout de la notification explicite d'encombrement (ECN) à IP", septembre 2001. *(P.S.)*

Appendice A – Segments dupliqués

Dans l'environnement actuel (sans notification explicite d'encombrement [Flo94], [RFC2481]) toutes les mises en œuvre de TCP utilisent les abandons de segment comme des indications de la part du réseau sur les limites de la bande passante disponible. On explique ici que le changement pour une plus grande fenêtre initiale ne devrait pas résulter en ce que l'envoyeur retransmette un grand nombre de segments dupliqués qui sont déjà arrivés chez le receveur.

Si un segment est abandonné dans la fenêtre initiale, il y a trois façons différentes pour que TCP récupère : (1) le démarrage lent à partir d'une fenêtre de un segment, comme on le fait après une fin de temporisation de retransmission, ou après une retransmission rapide dans TCP Tahoe ; (2) la récupération rapide sans accusé de réception sélectif (SACK, *selective acknowledgment*) comme on le fait après trois ACK dupliqués dans TCP Reno ; et (3) la récupération rapide avec SACK, pour TCP où l'envoyeur et le receveur prennent tous deux en charge l'option SACK [FF96]. Dans les trois cas, si un seul segment est abandonné de la fenêtre initiale, aucun segment dupliqué (c'est-à-dire, des segments qui ont déjà été reçus par le receveur) n'est transmis. Noter que pour une mise en œuvre de TCP qui envoie quatre segments de 512 octets dans la fenêtre initiale, un seul abandon de segment ne va pas exiger une fin de temporisation de retransmission, mais peut être récupéré en utilisant l'algorithme de retransmission rapide (sauf si le temporisateur de retransmission arrive à expiration prématurément). De plus, un seul segment abandonné d'une fenêtre initiale de trois segments peut être réparée en utilisant l'algorithme de retransmission rapide, selon le segment qui est abandonné et si des ACK retardés sont ou non utilisés. Par exemple, abandonner le premier segment d'une fenêtre initiale de trois segments va toujours exiger d'attendre une fin de temporisation, en l'absence de transmission limitée [RFC3042]. Cependant, abandonner le troisième segment va toujours permettre la récupération via l'algorithme de retransmission rapide, tant qu'aucun ACK n'est perdu.

On examine ensuite les scénarios où la fenêtre initiale contient deux à quatre segments, et où au moins deux de ces segments sont abandonnés. Si tous les segments dans la fenêtre initiale sont abandonnés, il est clair qu'alors aucun segment dupliqué n'est retransmis, car le receveur n'a pas encore reçu de segment. (Il y a encore une possibilité que ces segments abandonnés aient utilisé une bande passante raréfiée sur le chemin de leur point d'abandon ; cette question a été discutée à la Section 5.)

Lorsque deux segments sont abandonnés d'une fenêtre initiale de trois segments, l'envoyeur va seulement envoyer un segment dupliqué si les deux premiers des trois segments ont été abandonnés, et si l'envoyeur ne reçoit pas un paquet avec l'option SACK pour accuser réception du troisième segment.

Lorsque deux segments sont abandonnés d'une fenêtre initiale de quatre segments, un examen des six scénarios possibles (qu'on ne va pas reprendre ici) montre que, selon la position des paquets abandonnés, en l'absence de SACK, l'envoyeur peut envoyer un segment dupliqué. Il n'y a pas de scénario dans lequel l'envoyeur envoie deux segments dupliqués.

Lorsque trois segments sont abandonnés d'une fenêtre initiale de quatre segments, alors, en l'absence de SACK, il est possible qu'un segment dupliqué soit envoyé, selon la position des segments abandonnés.

Pour résumer, en l'absence de SACK, il y a des scénarios avec plusieurs abandons de segment de la fenêtre initiale où un segment dupliqué sera transmis. Il n'y a pas de scénario dans lequel plus d'un segment dupliqué soit transmis. Notre conclusion est que le nombre de segments dupliqués transmis par suite d'une plus grande fenêtre initiale devrait être faible.

Adresse des auteurs

Mark Allman
BBN Technologies/NASA Glenn Research Center
21000 Brookpark Rd
MS 54-5
Cleveland, OH 44135
mél: mallman@bbn.com
http://roland.lerc.nasa.gov/~mallman/

Sally Floyd
ICSI Center for Internet Research
1947 Center St, Suite 600
Berkeley, CA 94704
téléphone : +1 (510) 666-2989
mél : floyd@icir.org
http://www.icir.org/floyd/

Craig Partridge
BBN Technologies
10 Moulton St
Cambridge, MA 02138
mél : craig@bbn.com

Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'édition des RFC est actuellement fourni par l'Internet Society.