

Groupe de travail Réseau  
**Request for Comments : 3042**  
Catégorie : En cours de normalisation  
Traduction Claude Brière de L'Isle

M. Allman, NASA GRC/BBN  
H. Balakrishnan, MIT  
S. Floyd, ACIRI  
janvier 2001

## Amélioration de la récupération de perte dans TCP avec la transmission limitée

### Statut de ce mémoire

Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et des suggestions pour son amélioration. Prière de se reporter à l'édition actuelle du STD 1 "Normes des protocoles officiels de l'Internet" pour connaître l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

### Notice de copyright

Copyright (C) The Internet Society (2001). Tous droits réservés.

### Résumé

Le présent document propose un nouveau mécanisme du protocole de contrôle de transmission (TCP, *Transmission Control Protocol*) qui peut être utilisé pour récupérer plus efficacement les segments perdus lorsque la fenêtre d'encombrement d'une connexion est petite, ou lorsque un grand nombre de segments sont perdus dans une seule fenêtre de transmission. L'algorithme "Transmission limitée" invoque l'envoi d'un nouveau segment de données en réponse à chacun des deux premiers accusés de réception dupliqués qui arrivent chez l'expéditeur. La transmission de ces segments augmente la probabilité que TCP puisse récupérer d'un seul segment perdu en utilisant l'algorithme de retransmission rapide, plutôt que d'utiliser une coûteuse temporisation de retransmission. La transmission limitée peut être utilisée aussi bien en conjonction avec le, et en l'absence du, mécanisme TCP d'accusé de réception sélectif (SACK).

## 1. Introduction

Un certain nombre de chercheurs ont observé que les stratégies de récupération de perte de TCP ne fonctionnent pas bien lorsque la fenêtre d'encombrement est petite chez un expéditeur TCP. Cela peut arriver, par exemple, parce qu'il y a seulement une quantité limitée de données à envoyer, ou à cause de la limite imposée par la fenêtre annoncée par le receveur, ou à cause des contraintes imposées par le contrôle d'encombrement de bout en bout sur une connexion avec un petit produit bande passante-délai [Riz96], [Mor97], [BPS+98], [Bal98], [LK98]. Lorsque un TCP détecte un segment manquant, il entre dans une phase de récupération de perte en utilisant une des deux méthodes suivantes.

D'abord, si un accusé de réception (ACK) n'est pas reçu pour un certain segment dans un certain délai, une fin de temporisation de retransmission intervient et le segment est envoyé à nouveau [RFC0793], [PA00]. La seconde; l'algorithme "Retransmission rapide" envoie à nouveau un segment lorsque trois ACK dupliqués arrivent chez l'expéditeur [Jac88], [RFC2581]. Cependant, comme les ACK dupliqués provenant du receveur sont aussi déclenchés par le déclassement des paquets dans l'Internet, l'expéditeur TCP attend qu'il y ait trois ACK dupliqués pour tenter de surmonter l'ambiguïté entre perte de segment et désordre des paquets. Une fois qu'on est dans une phase de récupération de perte, un certain nombre de techniques peuvent être utilisées pour retransmettre les segments perdus, incluant la récupération fondée sur le démarrage lent ou la récupération rapide [RFC2581], NewReno [RFC2582], et la récupération de perte fondée sur les accusés de réception sélectifs (SACK) [RFC2018], [FF96].

La temporisation de retransmission (RTO, *retransmission timeout*) de TCP se fonde sur le temps d'aller-retour (RTT, *round-trip time*) mesuré entre l'expéditeur et le receveur, comme spécifié dans la [RFC2988]. Pour prévenir les retransmissions parasites de segments qui sont seulement retardés et non perdus, le RTO minimum est prudemment choisi comme étant d'une seconde. Donc, il incombe aux expéditeurs TCP de détecter et récupérer d'autant de pertes que possible sans subir une interminable temporisation lorsque la connexion reste en repos. Cependant, si il n'arrive pas du receveur suffisamment d'ACK dupliqués, l'algorithme de retransmission rapide n'est jamais déclenché --- cette situation survient lorsque la fenêtre d'encombrement est petite ou si un grand nombre de segments dans une fenêtre sont perdus. Par exemple, considérons une fenêtre d'encombrement (cwnd) de trois segments. Si un segment est éliminé par le réseau, au plus deux ACK dupliqués vont alors arriver à l'expéditeur. Comme trois ACK dupliqués sont nécessaires pour déclencher la retransmission rapide, une temporisation sera nécessaire pour envoyer à nouveau le paquet éliminé.

[BPS+97] a trouvé qu'environ 56 % des retransmissions envoyées par un serveur de la Toile actif étaient envoyés après l'arrivée à expiration du RTO, alors que seulement 44 % étaient traités par la retransmission rapide. De plus, seulement 4 % des retransmissions fondées sur le RTO auraient pu être évitées avec SACK, ce qui, bien sûr, doit continuer à lever les

ambiguïtés entre déclassement et perte authentique. À l'opposé, à utiliser la technique présentée dans le présent document et dans [Bal98], 25% des retransmissions fondées sur RTO dans cet ensemble de données auraient vraisemblablement été évitées.

La section suivante du présent document met en évidence les petits changements pour les envoyeurs TCP qui vont diminuer la dépendance au temporisateur de retransmission, et donc améliorer les performances de TCP lorsque une retransmission rapide n'est pas déclenchée. Ces changements n'ont pas d'effet néfaste sur les performances de TCP ni n'interagissent défavorablement avec les autres connexions, dans d'autres circonstances.

## 1.1 Terminologie

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" dans ce document sont à interpréter comme décrit dans la [RFC2119] et indiquent les niveaux d'exigence pour les protocoles.

## 2. Algorithme de transmission limitée

Lorsque un envoyeur TCP a en file d'attente pour transmission des données non envoyées précédemment, il DEVRAIT utiliser l'algorithme Transmission limitée, qui invite un envoyeur TCP à transmettre de nouvelles données à l'arrivée du premier de deux ACK dupliqués consécutifs lorsque les conditions suivantes sont satisfaites:

- \* La fenêtre annoncée du receveur permet la transmission du segment.
- \* Si la quantité de données en cours va rester inférieure ou égale à la fenêtre d'encombrement plus deux segments. En d'autres termes, l'envoyeur peut seulement envoyer deux segments au delà de la fenêtre d'encombrement (cwnd).

La fenêtre d'encombrement (cwnd) NE DOIT PAS être changée lorsque ces nouveaux segments sont transmis. En supposant que ces nouveaux segments et les ACK correspondants ne soient pas abandonnés, cette procédure permet à l'envoyeur de déduire la perte en utilisant le seuil standard de récupération rapide de trois ACK dupliqués [RFC2581]. Ceci est plus robuste pour les paquets déclassés que si un vieux paquet était retransmis au premier ou second ACK dupliqué.

Note : Si la connexion utilise des accusés de réception sélectifs [RFC2018], l'envoyeur des données NE DOIT PAS envoyer de nouveaux segments en réponse aux ACK dupliqués qui ne contiennent pas de nouvelles informations de SACK, car un receveur qui se comporte mal pourrait générer de tels ACK pour déclencher la transmission inappropriée de segments de données. Voir dans [SCWA99] un exposé sur les attaques par des receveurs qui se conduisent mal.

La transmission limitée suit le principe de contrôle d'encombrement de "conservation des paquets" [Jac88]. Chacun des deux premiers ACK dupliqués indique qu'un segment a quitté le réseau. De plus, l'envoyeur n'a pas encore décidé qu'un segment a été éliminé et donc n'a pas de raisons de supposer que l'état actuel de contrôle d'encombrement est inapproprié. Donc, transmettre des segments ne s'écarte pas de l'esprit des principes de contrôle d'encombrement de TCP.

[BPS99] montre que le déclassement des paquets n'est pas un événement rare sur le réseau. La [RFC2581] ne traite pas de l'envoi de données sur les deux premiers ACK dupliqués qui arrivent chez l'envoyeur. Cela cause l'envoi d'une salve de segments lorsque un ACK pour de nouvelles données arrive à la suite d'un reclassement des paquets. En utilisant la transmission limitée, les paquets de données seront rythmés par les ACK entrants, et donc la transmission ne sera pas aussi saccadée.

Note: La transmission limitée est mise en œuvre dans le simulateur ns [NS]. Les chercheurs qui souhaitent mieux étudier ce mécanisme peuvent le faire en activant "singledup\_" pour la connexion TCP concernée.

## 3. Travaux connexes

Le déploiement de la notification explicite d'encombrement (ECN, *Explicit Congestion Notification*) [Flo94], [RFC2481] peut bénéficier de connexions avec de petites tailles de fenêtre d'encombrement [RFC2884]. ECN fournit une méthode pour indiquer l'encombrement à l'hôte distant sans éliminer de segments. Bien que certains abandons de segment puissent quand même survenir, ECN peut permettre à TCP de mieux fonctionner avec de petites tailles de fenêtre d'encombrement

parce que l'expéditeur peut éviter beaucoup des temporisations de retransmission et des retransmissions rapides qui auraient autrement été nécessaires pour détecter les segments éliminés [RFC2884].

Lorsque le trafic TCP à capacité ECN est en compétition avec du trafic TCP sans capacité ECN, le trafic à capacité ECN peut recevoir jusqu'à 30 % de débit en plus. Pour les transferts en vrac, le bénéfice de performances relatif de ECN est supérieur lorsque en moyenne chaque flux a 3-4 paquets en cours durant chaque durée d'aller-retour [ZQ00]. Cela pourrait être une bonne estimation de l'impact sur les performances d'un flux utilisant la transmission limitée, car ECN et la transmission limitée réduisent toutes deux le besoin de s'appuyer sur le temporisateur de retransmission pour signaler l'encombrement.

L'algorithme de contrôle d'encombrement par réduction du débit [MSML99] utilise une forme de transmission limitée, car il invite à la transmission d'un segment de données toutes les fois qu'un second ACK dupliqué arrive chez l'expéditeur. L'algorithme découple la décision sur quoi envoyer de la décision de quand envoyer. Cependant, comme pour la transmission limitée, l'algorithme va toujours envoyer un nouveau segment de données sur le second ACK dupliqué qui arrive chez l'expéditeur.

#### 4. Considérations pour la sécurité

Les implications supplémentaires pour la sécurité des changements proposés dans le présent document, comparées à la vulnérabilité actuelle de TCP, sont minimales. Les problèmes de sécurité potentiels viennent de la subversion du contrôle d'encombrement de bout en bout provenant de "faux" ACK dupliqués, où un "faux" ACK dupliqué est un ACK dupliqué qui en fait n'accuse pas réception de nouvelles données reçues chez le receveur TCP. De faux ACK dupliqués pourraient résulter d'ACK dupliqués qui sont eux-mêmes dupliqués dans le réseau par des receveurs TCP qui se conduisent mal et envoient de faux ACK dupliqués pour subvertir le contrôle d'encombrement de bout en bout [SCWA99], [RFC2581].

Lorsque le receveur de données TCP a accepté d'utiliser l'option SACK, l'expéditeur de données TCP a une très bonne protection contre les faux ACK dupliqués. En particulier, avec SACK, un ACK dupliqué qui accuse réception de nouvelles données arrivant chez le receveur rapporte les numéros de séquence de ces nouvelles données. Donc, avec SACK, l'expéditeur TCP peut vérifier qu'un ACK dupliqué arrivant accuse réception de données que l'expéditeur TCP a bien envoyées, et pour lesquelles aucun accusé de réception précédent n'avait été reçu, avant d'envoyer de nouvelles données suite à cet accusé de réception. Pour plus de protection, l'expéditeur TCP pourrait garder un enregistrement des frontières de paquet pour les paquets de données transmis, et reconnaître au plus un accusé de réception valide pour chaque paquet (par exemple, le premier accusé de réception acquitte la réception de tous les numéros de séquence dans ce paquet).

On pourrait imaginer une protection limitée contre les faux ACK dupliqués pour une connexion TCP non SACK, où l'expéditeur TCP garde un enregistrement du nombre de paquets transmis, et reconnaît au plus un accusé de réception par paquet à utiliser pour déclencher l'envoi de nouvelles données. Cependant, cette comptabilité des paquets transmis et acquittés exigerait un état supplémentaire et ajouterait de la complexité chez l'expéditeur TCP, et ne semble pas nécessaire.

La protection la plus importante contre les faux ACK dupliqués vient du potentiel limité d'ACK dupliqués pour subvertir le contrôle d'encombrement de bout en bout. Il y a deux cas distincts à considérer : lorsque l'expéditeur TCP reçoit moins qu'un nombre seuil d'ACK dupliqués, et lorsque l'expéditeur TCP reçoit au moins un nombre seuil d'ACK dupliqués.

Dans le premier cas, un TCP avec transmission limitée va se comporter essentiellement de la même façon qu'un TCP sans transmission limitée parce que la fenêtre d'encombrement sera divisée par deux et qu'une période de récupération de pertes sera initiée.

Lorsque un expéditeur TCP reçoit moins que le nombre seuil d'ACK dupliqués, un receveur qui se conduit mal pourrait envoyer deux ACK dupliqués après chaque ACK régulier. On pourrait imaginer que l'expéditeur TCP enverrait à trois reprises à son taux d'envoi permis. Cependant, en utilisant la transmission limitée comme exposé à la section 2, l'expéditeur n'est autorisé qu'à excéder la fenêtre d'encombrement de moins que le seuil d'ACK dupliqués (de trois segments) et donc n'enverrait pas un nouveau paquet pour chaque ACK dupliqué reçu.

#### Remerciements

Bill Fenner, Jamshid Mahdavi et le groupe de travail Transport Area ont fourni de précieuses réactions sur la première version du présent document.

**Références**

- [Bal98] Hari Balakrishnan. "Challenges to Reliable Data Transport over Heterogeneous Wireless Networks". Thèse de doctorat, University of California at Berkeley, août 1998.
- [BPS+97] Hari Balakrishnan, Venkata Padmanabhan, Srinivasan Seshan, Mark Stemm, and Randy Katz. "TCP Behavior of a Busy Web Server: Analysis and Improvements". Rapport technique UCB/CSD-97-966, août 1997. Disponible à <http://nms.lcs.mit.edu/~hari/papers/csd-97-966.ps>. (Aussi dans Proc. IEEE INFOCOM Conf., San Francisco, CA, mars 1998.)
- [BPS99] Jon Bennett, Craig Partridge, Nicholas Shectman. "Packet Reordering is Not Pathological Network Behavior". IEEE/ACM Transactions on Networking, décembre 1999.
- [FF96] Kevin Fall, Sally Floyd. "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP". ACM Computer Communication Review, juillet 1996.
- [Flo94] Sally Floyd. "TCP and Explicit Congestion Notification". ACM Computer Communication Review, octobre 1994.
- [Jac88] Van Jacobson. "Congestion Avoidance and Control". ACM SIGCOMM 1988.
- [LK98] Dong Lin, H.T. Kung. "TCP Fast Recovery Strategies: Analysis and Improvements". Proceedings of InfoCom, mars 1998.
- [MSML99] Matt Mathis, Jeff Semke, Jamshid Mahdavi, Kevin Lahey. "The Rate Halving Algorithm", 1999. URL: [http://www.psc.edu/networking/rate\\_halving.html](http://www.psc.edu/networking/rate_halving.html) .
- [Mor97] Robert Morris. "TCP Behavior with Many Flows". Proceedings of the Fifth IEEE International Conference on Network Protocols. octobre 1997.
- [NS] "Ns network simulator". URL: <http://www.isi.edu/nsnam/> .
- [RFC0793] J. Postel (éd.), "Protocole de [commande de transmission](#) – Spécification du protocole du programme Internet DARPA", STD 7, septembre 1981.
- [RFC2018] M. Mathis et autres, "Options d'[accusé de réception sélectif](#) sur TCP", octobre 1996. (P.S.)
- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.
- [RFC2481] K. Ramakrishnan, S. Floyd, "Proposition d'ajout de la [notification d'encombrement explicite](#) (ECN) à IP", janvier 1999.
- [RFC2581] M. Allman, V. Paxson et W. Stevens, "[Contrôle d'encombrement avec TCP](#)", avril 1999. (Obsolète, voir RFC5681)
- [RFC2582] S. Floyd, T. Henderson, "[Modification NewReno à l'algorithme](#) de récupération rapide de TCP", avril 1999. (Obsolète, voir RFC3782) (Expérimentale)
- [RFC2884] J. Hadi Salim et U. Ahmed, "Évaluation des performances de la notification d'encombrement explicite (ECN) dans les réseaux IP", juillet 2000. (Information)
- [RFC2988] V. Paxson, M. Allman, "[Calcul du temporisateur](#) de retransmission de TCP", novembre 2000. (P.S. remplacée par la RFC6298)
- [Riz96] Luigi Rizzo. "Issues in the Implementation of Selective Acknowledgments for TCP". janvier 1996. URL : <http://www.iet.unipi.it/~luigi/selack.ps>
- [SCWA99] Stefan Savage, Neal Cardwell, David Wetherall, Tom Anderson. "TCP Congestion Control with a Misbehaving Receiver". ACM Computer Communications Review, octobre 1999.
- [ZQ00] Yin Zhang and Lili Qiu, "Understanding the End-to-End Performance Impact of RED in a Heterogeneous Environment", Cornell CS Technical Report 2000-1802, juillet 2000. URL <http://www.cs.cornell.edu/yzhang/papers.htm> .

**Adresse des auteurs**

Mark Allman  
NASA Glenn Research Center  
Lewis Field  
21000 Brookpark Rd. MS 54-5  
Cleveland, OH 44135  
téléphone : +1-216-433-6586  
Fax : +1-216-433-8705  
mél : mallman@grc.nasa.gov  
<http://roland.grc.nasa.gov/~mallman>

Hari Balakrishnan  
Laboratory for Computer Science  
545 Technology Square  
MIT  
Cambridge, MA 02139  
mél : [hari@lcs.mit.edu](mailto:hari@lcs.mit.edu)  
<http://nms.lcs.mit.edu/~hari/>

Sally Floyd  
AT&T Center for Internet Research at ICSI  
1947 Center St, Suite 600  
Berkeley, CA 94704  
téléphone : +1-510-666-2989  
mél : [floyd@aciri.org](mailto:floyd@aciri.org)  
<http://www.aciri.org/floyd/>

**Déclaration complète de droits de reproduction**

Copyright (C) The Internet Society (2001).

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et à [www.rfc-editor.org](http://www.rfc-editor.org), et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations y contenues sont fournies sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations ci encloses ne violent aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

**Propriété intellectuelle**

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr>.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

**Remerciement**

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