Internet Engineering Task Force (IETF)
Request for Comments : 6298
RFC rendue obsolete : 2988
RFC mise à jour : 1122
Catégorie : En cours de normalisation
ISSN : 2070-1721

V. Paxson, ICSI/UC Berkeley
M. Allman, ICSI
J. Chu, Google
M. Sargent, CWRU
juin 2011
Traduction Claude Brière de L'Isle

Calcul du temporisateur de retransmission de TCP

Résumé
Le présent document définit l'algorithme standard que les envoyeurs du protocole de contrôle de transmission (TCP, *Transmission Control Protocol*) sont requis d'utiliser pour calculer et gérer leur temporisateur de retransmission. Il s'étend sur la discussion du paragraphe 4.2.3.1 de la RFC1122 et relève l'exigence de prise en charge de l'algorithme de DEVRAIT à DOIT. Le présent document rend obsolète la RFC2988.

Statut de ce mémoire
Ceci est un document de l'Internet en cours de normalisation.

Le présent document a été produit par l'équipe d'ingénierie de l'Internet (IETF). Il représente le consensus de la communauté de l'IETF. Il a subi une révision publique et sa publication a été approuvée par le groupe de pilotage de l'ingénierie de l'Internet (IESG). Tous les documents approuvés par l'IESG ne sont pas candidats à devenir une norme de l'Internet ; voir la Section 2 de la RFC5741.

Les informations sur le statut actuel du présent document, tout errata, et comment fournir des réactions sur lui peuvent être obtenues à http://www.rfc-editor.org/info/rfc6298

Notice de droits de reproduction
Copyright (c) 2011 IETF Trust et les personnes identifiée comme auteurs du document. Tous droits réservés.

Le présent document est soumis au BCP 78 et aux dispositions légales de l'IETF Trust qui se rapportent aux documents de l'IETF (http://trustee.ietf.org/license-info) en vigueur à la date de publication de ce document. Prière de revoir ces documents avec attention, car ils décrivent vos droits et obligations par rapport à ce document. Les composants de code extraits du présent document doivent inclure le texte de licence simplifié de BSD comme décrit au paragraphe 4.e des dispositions légales du Trust et sont fournis sans garantie comme décrit dans la licence de BSD simplifiée.

1. Introduction

Le protocole de contrôle de transmission (TCP, *Transmission Control Protocol*) [RFC0793] utilise un temporisateur de retransmission pour assurer la livraison des données en l'absence de retour de la part du receveur distant des données. La durée de ce temporisateur est appelée la temporisation de retransmission (RTO, *retransmission timeout*). La [RFC1122] spécifie que la RTO devrait être calculée comme exposé dans [Jac88].

Le présent document codifie l'algorithme pour établir la RTO. De plus, le présent document s'étend sur la discussion du paragraphe 4.2.3.1 de la RFC1122 et relève l'exigence de prise en charge de l'algorithme de DEVRAIT à DOIT. La [RFC5681] esquisse l'algorithme que TCP utilise pour commencer l'envoi après l'arrivée à expiration de la RTO et pour envoyer une retransmission. Le présent document n'altère pas le comportement esquissé par la [RFC5681].

Dans certaines situations, il peut être bénéfique pour un envoyeur TCP d'être plus prudent que ce que les algorithmes détaillés dans le présent document permettent. Cependant, un TCP NE DOIT PAS être plus agressif que ce que les algorithmes qui suivent permettent. Le présent document rend obsolète la [RFC2988].

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].

2. Algorithme de base

Pour calculer la RTO actuelle, un envoyeur TCP entretient deux variables d'état, le délai d'aller-retour lissé (SRTT, *smoothed round-trip time*) et la variation de délai d'aller-retour (RTTVAR, *round-trip time variation*). De plus, on suppose une granularité d'horloge de G secondes.

Les règles qui gouvernent le calcul de SRTT, RTTVAR, et de la RTO sont les suivantes :

(2.1) Jusqu'à ce qu'une mesure du délai d'aller-retour (RTT) ait été faite pour un segment envoyé entre l'envoyeur et le receveur, l'envoyeur DEVRAIT établir la RTO inférieure à 1 seconde, bien que le "repli" sur les retransmissions répétées discuté en (5.5) s'applique quand même.

Noter que la version précédente du présent document utilisait une RTO initiale de 3 secondes [RFC2988]. Une mise en œuvre de TCP PEUT toujours utiliser cette valeur (ou toute autre valeur> 1 seconde). Ce changement de la limite inférieure de la RTO initiale est discuté plus en détails à l'Appendice A.

(2.2) Lorsque la mesure du premier RTT est faite, l'hôte DOIT régler /
SRTT < R
RTTVAR < R/2
RTO < SRTT + max (G, K*RTTVAR)
où K = 4.

(2.3) Lorsque une mesure suivante R' du RTT est faite, un hôte DOIT régler :
RTTVAR < (1 - bêta) * RTTVAR + bêta * |SRTT - R'|
SRTT < (1 - alpha) * SRTT + alpha * R'

La valeur de SRTT utilisée dans la mise à jour de RTTVAR est sa valeur avant la mise à jour de SRTT lui-même en utilisant la seconde allocation. C'est-à-dire que la mise à jour de RTTVAR et SRTT DOIT être calculée dans l'ordre ci-dessus.
Ce qui est ci-dessus DEVRAIT être calculé en utilisant alpha = 1/8 et bêta = 1/4 (comme suggéré dans [JK88]).
Après le calcul, un hôte DOIT mettre à jour RTO < SRTT + max (G, K*RTTVAR)

(2.4) Chaque fois que le RTO est calculé, si il est inférieur à 1 seconde, alors le RTO DEVRAIT être arrondi à 1 seconde. Traditionnellement, les mises en œuvre de TCP utilisent des horloges d'une granularité grossière pour mesurer le RTT et déclencher la RTO, ce qui impose une grande valeur minimum au RTO. Les recherches suggèrent qu'une grande RTO minimum est nécessaire pour que TCP reste prudent et évite les retransmissions parasites [AP99]. Donc, la présente spécification exige une grande RTO minimum comme approche prudente, tout en reconnaissant en même temps qu'à l'avenir, les recherches pourraient montrer qu'une RTO minimum plus petite est acceptable ou supérieure.

(2.5) Une valeur maximum PEUT être fixée pour la RTO pourvu qu'elle soit au moins de 60 secondes.

3. Prélèvement d'échantillons de RTT

TCP DOIT utiliser l'algorithme de Karn [KP87] pour prendre des échantillons de RTT. C'est-à-dire que les échantillons NE DOIVENT PAS être faits en utilisant des segments qui ont été retransmis (et donc pour lesquels on ne sait pas si la réponse était faite pour la première instance du paquet ou pour une instance ultérieure). Le seul cas où TCP peut en toute sécurité prendre des échantillons de RTT à partir de segments retransmis est lorsque on emploie l'option Horodatages [RFC1323], car l'option Horodatages ôte l'ambiguïté concernant l'instance du segment de données qui a déclenché l'accusé de réception.

Traditionnellement, les mises en œuvre de TCP ont pris une mesure de RTT à un instant (normalement, une fois par RTT). Cependant, lorsque on utilise l'option Horodatages, chaque ACK peut être utilisé comme échantillon de RTT. La [RFC1323] suggère que les connexions TCP qui utilisent de grandes fenêtres d'encombrement devraient prendre de nombreux échantillons de RTT par fenêtre de données pour éviter les effets d'alias dans le RTT estimé. Une mise en œuvre de TCP DOIT prendre au moins une mesure de RTT par RTT (sauf si cela n'est pas possible selon l'algorithme de Karn).

Pour des tailles de fenêtre d'encombrement très modestes, les recherches suggèrent que mesurer chaque segment ne conduit pas à un meilleur estimateur de RTT [AP99]. De plus, lorsque plusieurs échantillons sont pris par RTT, l'alpha et le bêta définis à la Section 2 peuvent garder un historique de RTT inadéquat. Une méthode pour changer ces constantes est actuellement une question ouverte pour la recherche.

4. Granularité d'horloge

Il n'y a pas d'exigences sur la granularité d'horloge G pour le calcul des mesures de RTT et les différentes variables d'état. Cependant, si le terme K*RTTVAR dans le calcul de la RTO est égal à zéro, le terme de la variance DOIT être arrondi à G secondes (c'est-à-dire, utiliser l'équation donnée à l'étape 2.3).

RTO < SRTT + max (G, K*RTTVAR)

L'expérience a montré que les plus fines granularités d'horloge (≤ 100 ms) ont plutôt de meilleures performances que les granularités plus grossières.

Noter que [Jac88] expose plusieurs astuces qui peuvent être utilisées pour obtenir une meilleure précision de la part des temporisateurs à granularité grossière. Ces changements sont largement mis en œuvre dans les TCP actuels.

5. Gestion du temporisateur RTO

Une mise en œuvre DOIT gérer le ou les temporisateurs de retransmission d'une façon telle qu'un segment ne soit jamais retransmis trop tôt, c'est-à-dire, moins d'une RTO après la précédente transmission de ce segment.

L'algorithme suivant est RECOMMANDÉ pour la gestion du temporisateur de retransmission :

(5.1) Chaque fois qu'est envoyé un paquet contenant des données (y compris une retransmission) si le temporisateur ne fonctionne pas, le lancer afin qu'il arrive à expiration après RTO secondes (pour la valeur actuelle de la RTO).

(5.2) Quand toutes les données en instance ont été acquittées, arrêter le temporisateur de retransmission.

(5.3) Lorsque un ACK est reçu qui accuse réception de nouvelles données, relancer le temporisateur de retransmission afin qu'il arrive à expiration après RTO secondes (pour la valeur actuelle de la RTO).

Lorsque le temporisateur de retransmission expire, faire ce qui suit :

(5.4) Retransmettre le plus ancien segment qui n'a pas été acquitté par le receveur TCP.

(5.5) L'hôte DOIT régler RTO < RTO * 2 ("réduire le temporisateur"). La valeur maximum discutée en (2.5) ci-dessus peut être utilisée pour fournir une limite supérieure à cette opération de doublement.

(5.6) Lancer le temporisateur de retransmission, de façon qu'il arrive à expiration après RTO secondes (pour la valeur de RTO après l'opération de doublement mentionnée en 5.5).

(5.7) Si le temporisateur arrive à expiration en attendant le ACK d'un segment SYN et si la mise en œuvre de TCP utilise une RTO de moins de 3 secondes, la RTO DOIT être réinitialisée à 3 secondes lorsque commence la transmission des données (c'est-à-dire, après l'achèvement de la prise de contact en trois phases).

Cela représente un changement par rapport à la version précédente du présent document [RFC2988] et est discuté dans l'Appendice A.

Noter qu'après retransmission, une fois qu'une nouvelle mesure du RTT est obtenue (ce qui ne peut arriver que lorsque de nouvelles données ont été envoyées et acquittées) le calcul mentionné à la Section 2 est effectué, y compris le calcul de la RTO, qui peut résulter en un "écroulement" de la RTO après qu'elle a été soumise au repli exponentiel (règle 5.5).

Noter qu'une mise en œuvre de TCP PEUT supprimer SRTT et RTTVAR après plusieurs replis du temporisateur car il est vraisemblable que les SRTT et RTTVAR en cours sont erronés dans cette situation. Une fois que SRTT et RTTVAR sont supprimés, ils devraient être initialisés avec le prochain échantillon de RTT pris selon (2.2) plutôt qu'en utilisant (2.3).

6. Considérations pour la sécurité

Le présent document exige qu'un TCP attende pendant un certain intervalle de temps avant de retransmettre un segment non acquitté. Un attaquant pourrait faire qu'un envoyeur TCP calcule une grande valeur de RTO en ajoutant un délai à la latence d'un paquet bien cadencé, ou à son accusé de réception. Cependant, la capacité à ajouter un délai à la latence d'un paquet coïncide souvent avec la capacité à causer la perte du paquet, de sorte qu'il est difficile de voir ce que pourrait gagner un attaquant d'une telle attaque qui pourrait causer plus de dommages que de simplement éliminer certains des paquets de la connexion TCP.

L'Internet s'appuie, dans une mesure considérable, sur la mise en œuvre correcte de l'algorithme de la RTO (ainsi que de ceux décrits dans la RFC5681) afin de préserver la stabilité du réseau et d'éviter l'écroulement par encombrement. Un attaquant pourrait faire que les points d'extrémité TCP répondent de façon plus agressive en présence d'encombrement en falsifiant des accusés de réception pour des segments avant que le receveur ait réellement reçu les données, diminuant ainsi la RTO jusqu'à des valeurs non sûres. Mais faire ainsi exige de falsifier correctement les accusés de réception, ce qui est difficile sauf si l'attaquant peut surveiller le trafic le long du chemin entre l'envoyeur et le receveur. De plus, même si l'attaquant peut faire que la RTO de l'envoyeur atteigne une valeur trop petite, il apparaît que l'attaquant ne peut pas élever cela au niveau d'une attaque (comparé aux autres dommages qu'il pourrait causer si il peut falsifier les paquets qui appartiennent à la connexion) car l'envoyeur TCP va quand même réduire son temporisateur en présence de la perte d'un paquet incorrectement transmis, due à l'encombrement réel.

Les considérations pour la sécurité de la [RFC5681] sont aussi applicables au présent document.

7. Changements depuis la RFC2988

Le présent document réduit le RTO initial des 3 secondes précédentes RFC2988] à 1 seconde, sauf si le SYN ou le ACK du SYN est perdu, auquel cas le RTO par défaut revient à 3 secondes avant que commence la transmission des données.

8. Remerciements

L'algorithme de RTO décrit dans le présent mémoire a été généré par Van Jacobson in [Jac88].

Beaucoup des données qui motivent de changer le RTO initial de trois secondes à une sont venues de Robert Love, Andre Broido, et Mike Belshe.

9. Références

9.1 Références normatives

[RFC5681] M. Allman, V. Paxson, E. Blanton, "Contrôle d'encombrement de TCP", septembre 2009. *(D. S.)*

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

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

[RFC1323] V. Jacobson, R. Braden et D. Borman, "Extensions TCP pour de bonnes performances", mai 1992.

[RFC0793] J. Postel (éd.), "Protocole de commande de transmission – Spécification du protocole du programme Internet DARPA", STD 7, septembre 1981.

9.2 Références pour information

[AP99] Allman, M. and V. Paxson, "On Estimating End-to-End Network Path Properties", SIGCOMM 99.

[Chu09] Chu, J., "Tuning TCP Parameters for the 21st Century", http://www.ietf.org/proceedings/75/slides/tcpm-1.pdf , juillet 2009.

[HKA04] Henderson, T., Kotz, D., and Abyzov, I., "CRAWDAD trace dartmouth/campus/tcpdump/fall03 (v. 2004-11-09)", http://crawdad.cs.dartmouth.edu/dartmouth/campus/ tcpdump/fall03, novembre 2004.

[Jac88] Jacobson, V., "Congestion Avoidance and Control", Computer Communication Review, vol. 18, no. 4, pp. 314-329, août 1988.

[JK88] Jacobson, V. and M. Karels, "Congestion Avoidance and Control", ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z.

[KP87] Karn, P. and C. Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocols", SIGCOMM 87.

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

[SLS09] Schulman, A., Levin, D., and Spring, N., "CRAWDAD data set umd/sigcomm2008 (v. 2009-03-02)", http://crawdad.cs.dartmouth.edu/umd/sigcomm2008 , mars 2009.

Appendice A. Raison pour diminuer le RTO initial

Le choix d'une RTO initiale raisonnable requiert de faire la balance entre deux exigences contradictoires :

1. La RTO initiale devrait être suffisamment grande pour couvrir la plupart des chemins de bout en bout afin d'éviter les retransmissions parasites et l'impact négatif sur les performances qui leur est associé.
2. La RTO initiale devrait être assez petite pour assurer la récupération à temps d'une perte de paquet survenant avant qu'un échantillon de RTT soit prélevé.

Traditionnellement, TCP a utilisé 3 secondes comme RTO initiale [RFC1122], [RFC2988]. Le présent document invite à diminuer cette valeur à une seconde pour les raisons suivantes :

- Les réseaux modernes sont simplement plus rapides que ce qu'était l'état de l'art lorsque a été définie la RTO initiale de trois secondes.
- Les études ont montré que les temps d'aller-retour de plus de 97,5 % des connexions observées dans une analyse à grande échelle étaient de moins d'une seconde [Chu09], suggérant que 1 seconde satisfait au critère 1 ci-dessus.
- De plus, les études ont observé des taux de retransmission au sein de la prise de contact en trois phases d'en gros 2 %. Cela montre que réduire la RTO initiale est bénéfique à un ensemble non négligeable de connexions.
- Cependant, environ 2,5 % des connexions étudiée dans [Chu09] ont un RTT supérieur à une seconde. Pour ces connexions, une RTO initiale d'une seconde garantit une retransmission durant l'établissement de la connexion (nécessaire ou non).

Lorsque cela arrive, le présent document appelle à revenir à une RTO initiale de 3 secondes pou la phase de transmission des données. Donc, les implications de la retransmission parasite sont modestes : (1) un SYN supplémentaire est transmis dans le réseau, et (2) conformément à la [RFC5681] la fenêtre d'encombrement initiale sera limitée à un segment. Alors que (2) fait clairement subir un inconvénient à de telles connexions, le présent document règle au moins la RTO de telle façon que la connexion ne subisse pas des problèmes continuels avec une courte temporisation. (Bien sûr, si le RTT est de plus de 3 secondes, la connexion va quand même rencontrer des difficultés. Mais ce n'est pas un nouveau problème pour TCP.)

De plus, on note que lorsque on utilise les horodatages, TCP sera capable de prendre un échantillon de RTT même en présence d'une retransmission parasite, facilitant la convergence vers une estimation correcte du RTT lorsque celui-ci excède 1 seconde.

Comme preuve supplémentaire des résultats présentés dans [Chu09], on a analysé les traces de paquets de comportement de client collectées dans quatre terrains propices différents à des moments différents, comme suit :

Nom	**Dates**	**Paquets**	**Connexions**	**Clients**	**Serveurs**
LBL-1	Oct/05--Mar/06	292 M	242 k	228	74 k
LBL-2	Nov/09--Feb/10	1,1 G	1,2 M	1047	38 k
ICSI-1	Sep/11--18/07	137 M	2,1 M	193	486 k
ICSI-2	Sep/11--18/08	163 M	1,9 M	177	277 k
ICSI-3	Sep/14--21/09	334 M	3,1 M	170	253 k
ICSI-4	Sep/11--18/10	298 M	5 M	183	189 k
Dartmouth	Jan/4--21/04	1 G	4 M	3782	132 k
SIGCOMM	Aug/17--21/08	11,6 M	133 k	152	29 k

Les données "LBL" ont été prises au Lawrence Berkeley National Laboratory, les données "ICSI" au International Computer Science Institute, les données "SIGCOMM" au réseau sans fil qui desservait les participants au SIGCOMM 2008, et les données "Dartmouth" ont été collectée sur le réseau sans fil du Dartmouth College. Les deux derniers ensembles de données sont disponibles sur la bibliothèque de données CRAWDAD [HKA04], [SLS09]. Le tableau donne les dates de collecte des données, le nombre de paquets collectés, le nombre de connexions TCP observées, le nombre de clients locaux surveillés, et le nombre de serveurs distants contactée. *(k = kilo (10^3) M = Mega (10^6) G = Giga (10^9))*
On ne considère que les connexions initiée près du point de collecte.

L'analyse de ces ensembles de données montre que la prévalence des SYN retransmis st entre 0,03 % (ICSI-4) et à peu près 2 % (LBL-1 et Dartmouth).

On a ensuite analysé les données pour déterminer le nombre de retransmissions supplémentaires et parasites qui auraient été supportées si la RTO initiale avait été supposée de 1 seconde. Dans la plupart des ensembles de données, la proportion de connexions avec des retransmissions parasites était de moins de 0,1 %. Cependant, dans l'ensemble de données Dartmouth, approximativement 1,1 % des connexions auraient envoyé une retransmission parasite avec une RTO initiale inférieure. On attribue cela au fait que le réseau surveillé est sans fils et donc susceptible de retards supplémentaires provenant d'effets radiofréquences.

Finalement, il y a des avantages de performances évidents à retransmettre les SYN perdus avec une RTO initiale réduite. Sur nos ensembles de données, le pourcentage de connexions qui ont retransmis un SYN et auraient réalisé au moins une amélioration de performances de 10 % en utilisant la RTO initiale plus petite spécifiée dans ce document va de 43 % (LBL-1) à 87 % (ICSI-4). Le pourcentage de connexions qui auraient réalisé au moins une amélioration de performances de 50 % va de 17 % (ICSI-1 et SIGCOMM) à 73 % (ICSI-4).

À partir des données auxquelles nous avons eu accès, nous concluons que la RTO initiale inférieure est vraisemblablement bénéfique à de nombreuses connexions, et nuisibles pour relativement peu.

Adresse des auteurs

Vern Paxson
ICSI/UC Berkeley
1947 Center Street
Suite 600
Berkeley, CA 94704-1198
téléphone : 510-666-2882
mél : vern@icir.org
http://www.icir.org/vern/

Mark Allman
ICSI
1947 Center Street
Suite 600
Berkeley, CA 94704-1198
téléphone : 440-235-1792
mél : mallman@icir.org

Matt Sargent
Case Western Reserve Univ.
Olin Building
10900 Euclid Avenue
Room 505
Cleveland, OH 44106
téléphone : 440-223-5932
mél : mts71@case.edu

H.K. Jerry Chu
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
téléphone : 650-253-3010
mél : hkchu@google.com