
Comparaison des principaux algorithmes TCP

Thibaut Ehlinger, Université de Strasbourg

20 octobre 2015

TCP (*Transmission Control Protocol*) est une implémentation de la couche transport du modèle OSI. C'est de loin son implémentation la plus populaire, avec pour maigre concurrent UDP, qui est moins contraignant et donc plus utile pour les connexions ne nécessitant pas une grande robustesse. TCP étant omniprésent dans les réseaux actuels, il a donc souvent été étudié et réimplémenté, parfois parce qu'une implémentation présentait des faiblesses qui lui étaient inhérentes, parfois parce qu'aucune implémentation ne répondait à une problématique bien précise du réseau l'utilisant, comme c'est le cas pour les *long fat network*, par exemple. Dans cette rédaction, nous étudierons tout d'abord les différentes implémentations historiques de TCP. Celles ayant été utilisées un jour ou l'autre mais qui sont aujourd'hui obsolètes car peu efficaces. Puis nous étudierons certains des algorithmes utilisés aujourd'hui, dans le but de comprendre quels sont les critères déterminants lors du choix de l'algorithme utilisé. En effet ceux-ci peuvent-être plus ou moins "agressifs" ou stables.

Les premières implémentations de TCP

De TCP Tahoe à TCP Westwood, en passant par TCP Reno, nous allons vous présenter une série d'algorithmes qui se sont globalement succédés, chacun étant dans la plupart des cas plus efficace que son prédécesseur, en tout point.

Tahoe

Slow start & congestion avoidance

Cette implémentation est citée pour la première fois en 1988 dans un article de Van Jacobson et Karels. C'est la plus simple et la moins efficace, tout types de problème confondus. Au début, on est en **Slow Start** : tant qu'on n'a aucune perte, on double la taille de la *congestion window* (**CWND**), c'est à dire la taille en segments des rafales TCP envoyées. À chaque fois que tous les segments envoyés sont acquittés, on double la taille de la CWND. Le slow start est interrompu dès la première perte de paquet, c'est à dire lorsqu'on reçoit plusieurs fois le même acquittement (**ACK**) ou lors d'un **time out**.

Comme la plupart des pertes de paquets sur les réseaux filaires sont provoqués par des congestions, Tahoe implémente le mode **Congestion Avoidance (CA)**. On se base sur une variable nommée **ssthresh** (*slow-start threshold*). Lorsque $CWND \leq ssthresh$, on est en Slow Start, sinon on est en CA. En CA, on augmente la taille de la fenêtre de 1 segment à chaque fois que tous les segments d'une rafale sont acquittés. C'est donc une progression linéaire.

Fast retransmit

Si un ACK est reçu trois fois de suite, alors le segment (ACK+1) a probablement été perdu. Il ne s'agit probablement pas d'un déséquencement. Pour pallier ce problème, Tahoe renvoie directement le segment concerné au lieu d'attendre la fin d'un timeout. Après un fast retransmit, $ssthresh = \frac{CWND}{2}$ et $CWND = 1$

(Slow Start).

Avantages et faiblesses

En résumé, Tahoe est déjà une réelle avancée, car il permet d'approximer très grossièrement la plus grosse taille de fenêtre d'émission possible, et il reste stable à moins qu'il n'y ait des congestions sur le réseau. Dans ce cas, Tahoe peut s'adapter au problème et diminuer la taille de sa fenêtre d'émission.

Néanmoins, la réinitialisation de la fenêtre d'émission à 1 à la moindre perte de paquet est une sous-estimation beaucoup trop importante. La taille moyenne de CWND est donc bien plus faible que ce qu'elle pourrait être optimalement.

Reno

TCP résoud déjà grandement le problème de l'oscillation perpétuelle Slow Start/CA en ajoutant la notion de **Fast Recovery**.

Fast recovery

Après un fast retransmit, au lieu de passer en Slow Start on applique : $ssthresh = \frac{CWND}{2}$, $CWND = ssthresh + 3$. En fait on repasse directement en CA, tout en augmentant la taille de CWND de 3, en référence aux trois segments qui n'ont pas été reçus à cause des ACK dupliqués. En cas de time out néanmoins, on repasse en Slow Start. Cette stratégie permet en cas de perte de ne pas baisser drastiquement la taille de CWND, à moins d'un time out. Un time out est en fait plus grave qu'une simple perte de paquet. Il témoigne probablement d'une modification topologique du réseau ou bien d'une congestion importante, alors qu'une simple perte de paquet peut-être due à des événements plus ponctuels, tels qu'un déséquencement ou une perte.

Faiblesses

TCP Reno réagit bien aux pertes de paquets quand elles se limitent à une par rafale. Quand par contre il y en a plusieurs par rafale, Reno est presque aussi inefficace que Tahoe, car il ne peut détecter qu'une seule perte de paquet à la fois.

De plus, il se peut qu'au sein d'une même rafale, l'émetteur ait le temps de passer en Fast Recovery, puis à nouveau en Congestion Avoidance, **deux fois**. Ce qui veut dire que si la fenêtre est trop grande et les pertes trop espacées au sein de cette fenêtre, la CWND peut-être divisée par 4.

New Reno

New Reno peut détecter les pertes de paquets multiples et est, de fait, bien plus performant que Reno. En effet, New Reno garde une trace de tous les segments envoyés dans une rafale. Lorsqu'il détecte une perte, il **reste** en fast recovery tant que tous les segments de cette rafale n'ont pas été acquittés par le destinataire.

Problèmes

Puisque les pertes ne sont détectées que par les re-transmissions

Heading on level 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo.

First This is the first item

Last This is the last item

Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies