

Les variantes de TCP dans le cas de connexions antiparallèles

Tristan Braud, Martin Heusse

▶ **To cite this version:**

Tristan Braud, Martin Heusse. Les variantes de TCP dans le cas de connexions antiparallèles. ALGOTEL 2016 - 18èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, May 2016, Bayonne, France. hal-01305364

HAL Id: hal-01305364
https://hal.science/hal-01305364

Submitted on 20 Apr 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Les variantes de TCP dans le cas de connexions antiparallèles

Tristan Braud et Martin Heusse

Grenoble INP, Laboratoire d'Informatique de Grenoble CNRS UMR 5217
[Martin.Heusse, Tristan.Braud]@imag.fr

Lorsque deux connexions TCP antiparallèles partagent un goulot d'étranglement, leurs interactions sont souvent néfastes, avec de fortes oscillations des files d'attente d'une direction à l'autre. Dans le cas d'une ligne ADSL, où souvent le tampon montant est surdimensionné, l'*upload* nuit fortement au *download*, au détriment de services souvent contraints en délais comme la navigation *web*.

Nous montrons ici l'influence souvent significative de l'utilisation d'une variante ou l'autre de TCP, alors que les études précédentes considéraient uniquement *New Reno*. Les expériences portent tout d'abord sur deux connexions antiparallèles continues et ensuite sur le cas d'une charge aléatoire de connexions de tailles finies opposées à un *upload* ininterrompu.

Mots-clefs : algorithmes de contrôle de congestion ; lien asymétrique ; temps de réponse

1 Introduction

Deux connexions TCP sont antiparallèles quand elles partagent dans des directions opposées les mêmes goulots d'étranglement. On sait que les données présentes dans les files d'attente oscillent alors entre les tampons en tête des liens limitants. De ce fait, à tout instant, un seul des deux tampons peut être occupé continûment et une seule des connexions bénéficie entièrement de la capacité disponible — à l'exception des périodes de bascule où les données présentes dans un tampon se déversent dans la file d'attente opposée. Ce phénomène et ses conséquences ont été mis en évidence à plusieurs reprises pour des connexions TCP New Reno [1, 2] alors que d'autres variantes sont largement déployées. Or, ces dernières diffèrent au niveau de l'algorithme de contrôle de congestion qui joue un rôle central dans la dynamique du système.

Cet article se focalise sur les performances obtenues en situation réelle. Les variantes les plus populaires de l'algorithme (*New Reno*, *Vegas* [3], *Compound* [4] et *Cubic* [5]) sont opposées les unes aux autres sur divers liens asymétriques (où les conséquences de l'oscillation des données sont les plus marquées) de manière à déterminer leur sensibilité à la présence de telle ou telle variante en sens inverse.

2 Conditions expérimentales

Les expériences sont menées sur un banc d'essai composé de trois ordinateurs. Le routeur utilise FreeBSD 9.2, tandis que les hôtes alternent entre FreeBSD 9.2 pour les variantes *New Reno*, *Cubic* et *Vegas*, et Windows 7 pour TCP *Compound*. Sur le routeur, le lien asymétrique est obtenu à l'aide du module *Dummynet*. Ce dernier permet d'émuler sans artefact des liens jusqu'à 100Mb/s par l'utilisation d'une fréquence d'interruption supérieure à la normale.

	τ	C	Q
Download	42 ms	1.2MB/s (10Mb/s au niveau 2)	250 paquets
Upload	52 ms	240kB/s (2Mb/s au niveau 2)	40/250 paquets
TCP		*New Reno*, *Vegas*, *Compound*	

TABLE 1: Paramètres de l'expérience

Les conditions expérimentales sont décrites Table 1. Le lien descendant émulé a une capacité de 10Mb/s, tandis que le lien montant est à 2Mb/s. Les tampons montants font respectivement soit 40 paquets (un tampon correctement dimensionné, pour une latence raisonnable) soit 250 paquets (un tampon surdimensionné), tandis que que le tampon descendant est à 250 paquets. En effet, utiliser des tampons montant et descendant de tailles similaires est une pratique fréquente qui produit de fortes interférences entre *upload* et *download* [6]. Le lien ainsi émulé correspond à un lien ADSL typique, dont la capacité théorique maximale de 14Mb/s est rarement atteinte (ADSL2+ augmente le débit mais augmente aussi l'asymétrie).

3 Connexions longues

Les mesures sont effectuées pendant 600s, lorsque les deux connexions ont atteint le régime de *congestion avoidance*, 200 s après leur démarrage étagé aléatoirement. Pour obtenir des statistiques fiables et déterminer la variabilité des résultat, chaque expérience a été lancée 10 fois. Les résultats présentés correspondent aux valeurs moyennes et écarts types.

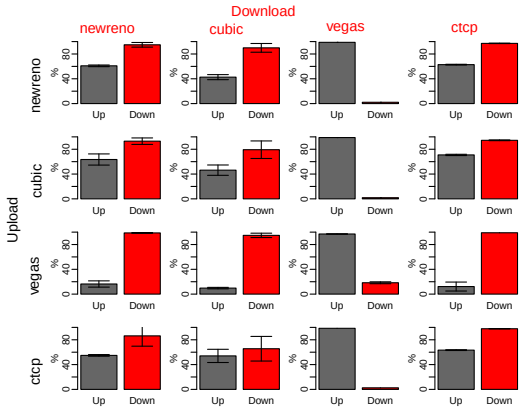


FIGURE 1: Utilisation du lien (%) et écart type pour un tampon correctement dimensionné — respectivement 250 et 40 paquets pour les liens montants et descendants — 10/2Mbps

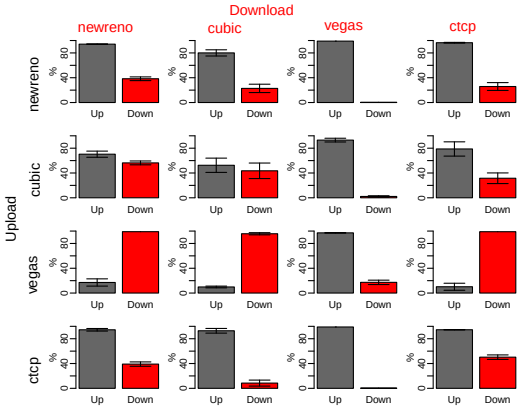


FIGURE 2: Utilisation du lien (%) et écart type pour un tampon montant surdimensionné — 250 paquets montants et descendants — 10/2Mbps

Les débits des connexions pour des tampons correctement dimensionnés (*i.e.* 40 paquets *vs.* 250 paquets) apparaissent Figure 1. Toutes les variantes à l'exception de *Vegas* atteignent la même utilisation du lien descendant, proche de 100%. En effet, un dimensionnement correct des tampons est le facteur clé dans le bon fonctionnement des algorithmes basés sur le cadencement des ACKs. La différence principale entre les versions de TCP émerge lorsque TCP *Cubic* est utilisé coté serveur, et *Compound* pour le client, c'est à dire dans le cas le plus fréquemment rencontré ! Cette combinaison donne le débit le plus faible dans le sens descendant, avec une variabilité élevée.

Bien que les tampons soient de tailles raisonnables, le produit délai-bande passante est suffisamment élevé pour déclencher les modes spécifiques de *Cubic* et *Compound*, dont le comportement diffère notablement de *New Reno*. *Cubic* en particulier devient ainsi plus agressif afin d'atteindre rapidement la capacité du lien et obtient des débits plus importants, au point parfois d'écraser la connexion montante (dont le débit diminue alors de 30 à 40%).

TCP *Vegas* maintient une occupation minimale de la file d'attente, ce qui s'avère désastreux : en général, le partage des ressources entre TCP *Vegas* et *New Reno*/*Cubic*/*Compound* se fait au détriment de TCP *Vegas*, qui s'efface devant les autres connexions [7]. Ce phénomène est bien connu pour deux transferts dans la même direction, cette expérience étend ce résultat au trafic sur la voie de retour. Incidemment, relancer l'expérience sur un lien asymétrique à 2.4Mb/s descendant pour 400Kb/s montant donne le même débit sur le lien montant car *Vegas* est ici en fait incapable de sonder la capacité disponible. Utiliser *Vegas* du côté serveur est tout simplement catastrophique pour le *download*.

Les RTT observés sont de l'ordre de 250ms (non représentés par manque d'espace), ce qui reste raisonnable pour la majorité des applications et correspond au fait que les *buffers* sont de tailles adéquates. L'agressivité plus importante de *Cubic* conduit à des RTT légèrement plus élevés (d'environs 10%). À l'opposé, *Vegas* parvient à garder l'occupation de la file d'attente au plus bas. Ainsi, les RTT observés sont bien moins élevés à la fois pour l'*upload* et le *download*.

La Figure 2 représente le pourcentage d'utilisation du lien obtenu en confrontant les variantes de TCP sur un lien asymétrique avec le même tampon aux deux extrémités du goulot d'étranglement. Dans cette configuration, les effet nocifs de la bascule sont très marqués avec TCP *New Reno* : l'*upload* atteint 95% de la capacité du lien, tandis que le *download* tombe aux alentours de 40%. Comme les délais augmentent dans le réseau, TCP *Cubic* passe plus de temps en "mode *Cubic*", et la différence entre *Cubic* et *New Reno* devient plus significative.

En contraste avec les autres variantes [6], TCP *Vegas* garde les tampons vides, le balancier de données ne peut donc avoir lieu. Toutefois, sa fenêtre de congestion est ajustée uniquement en fonction du RTT, ce qui a pour conséquence de faibles RTT, mais aussi des débits extrêmement réduits. Si TCP *Vegas* est inadapté à une utilisation sur le lien descendant, lorsqu'il est utilisé pour l'*upload*, le *download* en bénéficie fortement. En conclusion, *Vegas* est un très mauvais choix coté serveur, mais reste un bon candidat pour le client car il favorise la réactivité du téléchargement.

La composante basée sur le délai de *Compound* joue sur l'accélération de la croissance de la fenêtre. La connexion est donc plus agressive jusqu'à commencer à remplir le tampon, moment à partir duquel l'algorithme se comporte comme TCP *New Reno*. À l'instar de *Cubic*, cette agressivité ne bénéficie pas à Compound, qui présente des débits en *download* plus bas que *New Reno*.

Le fait d'avoir des tampons surdimensionnés dans le réseau mène à des RTT plus élevés, entre 600 et 800ms, sauf dans le cas de TCP *Vegas*. En effet, quelque soit la taille du tampon montant, *Vegas* ne le remplit pas, et les RTT observés sont identiques à ceux observés lors de l'expérience précédente.

4 Temps de réponse des téléchargements

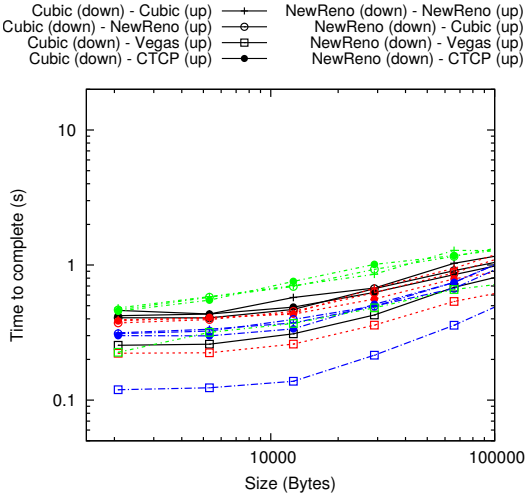


FIGURE 3: Temps de réponse conditionnels des connexions pour un trafic poissonien descendant et une connexion montante. Charge à 50%. Tampon correctement dimensionné : 10Mb/s - 250 pkts down 2Mb/s - 40 pkts up.

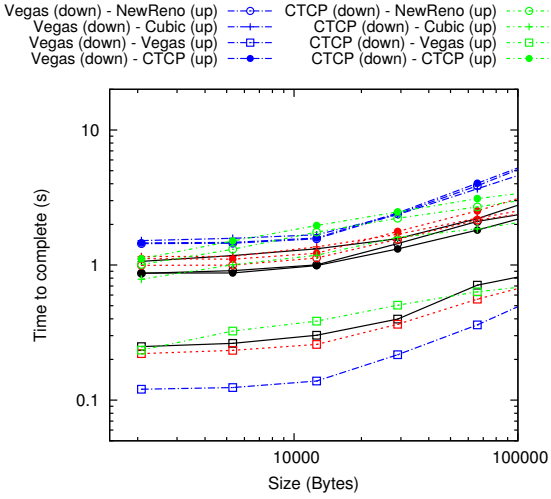


FIGURE 4: Temps de réponse conditionnels des connexions pour un trafic poissonien descendant et une connexion montante. Charge à 50%. Tampon surdimensionné : 10Mb/s - 250 pkts down 2Mb/s - 250 pkts up.

Afin de tester les diverses variantes de TCP dans un environnement plus réaliste, c'est à dire un trafic principalement constitué de petites connexions *web* et quelques transferts plus volumineux, cette section porte sur les interactions entre un *upload* long (1000 s) et une charge descendante complexe. La charge

descendante a été composée de connexions qui arrivent selon un processus de Poisson, tandis que leur taille suit une distribution à queue lourde. Le processus de Poisson a une intensité $\lambda = 0.15\,s^{-1}$, tandis que les tailles des transferts suivent une distribution de Zipf de paramètre $\alpha = 1.7$, ce qui correspond à une occupation descendante d'environs 50%.

L'objectif de cette expérience est de capturer le ralentissement qu'un utilisateur peut rencontrer lorsqu'il navigue sur Internet alors qu'un *upload* long (cloud, torrent, ftp) est actif.

En règle générale, les temps de réponse présentés dans les Figures 3 et 4 reflètent les résultats obtenus pour des connexions longues, avec des temps de réponse qui peuvent doubler pour des tampons surdimensionnés. Ainsi, lorsque *Vegas* est utilisé pour l'*upload* (points "carrés"), les temps de réponse en *download* sont les plus bas.

En revanche, la Figure 3 amène à des conclusions bien différentes de la Figure 1 : pour des transferts courts à moyens, c'est à dire lorsque le *slow start* influence encore les performances de manière significative, les temps de réponse atteints par *Vegas* sont sensiblement similaires à ceux des autres variantes, les temps de réponses les plus faibles étant obtenus dans le cas où *Vegas* est utilisé à la fois sur les liens montant et descendant ! En d'autres termes, dans un contexte réaliste, *Vegas* présente des performances supérieures à ses concurrents. En effet, lorsque *Vegas* est utilisé sur le lien montant, la taille du tampon a peu, voire aucun impact.

Curieusement, *Cubic* et *Compound* utilisés sur le lien descendant donnent des temps de réponse plus bas que *New Reno* (20%) et *Vegas* (25%) lorsque le tampon montant est surdimensionné, avec de très bons résultats lorsque *Compound* est utilisé sur le lien descendant et *Cubic* sur le lien montant (nous en profiterons pour savourer l'ironie de la chose, les statistiques montrent que la situation inverse est beaucoup plus plausible). Au contraire, lorsque ce tampon est bien dimensionné, TCP *Vegas* donne de très bons temps de réponse (50% meilleurs que *Compound*, et 20% inférieurs à *New Reno* et *Cubic*).

Par manque de place, nous mentionnerons simplement que ces deux expériences donnent des résultats sensiblement similaires avec une charge plus élevée (80%).

5 Conclusion

L'étude des variantes les plus populaires de TCP confirme et complète les études précédentes : l'effet de balancier est omniprésent, mais avec peu d'impact pour TCP *Vegas* qui ne sature pas le tampon montant et est donc insensible à son éventuel sur-dimensionnement.

Alors que les combinaisons de variantes TCP les plus attendues sont aussi les plus nuisibles, l'étude d'un trafic composé de connexions de taille finies remet *Vegas* sur l'échiquier, car il bénéficie alors du fort contraste entre la phase de *slow start* et *congestion avoidance* pour permettre au connexions courtes de se terminer rapidement.

Références

[1] Martin Heusse, Sears A Merritt, Timothy X Brown, and Andrzej Duda. Two-way TCP connections : old problem, new insight. *ACM SIGCOMM CCR*, 41, 2011.

[2] Denis Collange. Performance Model of Opposite TCP Connections on Asymmetric Capacities. In *SPECTS 2007*, 2007.

[3] Lawrence S. Brakmo, Sean W. O'Malley, and Larry L. Peterson. Tcp vegas : New techniques for congestion detection and avoidance. SIGCOMM '94, 1994.

[4] Kun Tan, Jingmin Song, Qian Zhang, and M. Sridharan. A compound tcp approach for high-speed and long distance networks. In *INFOCOM 2006*, 2006.

[5] Sangtae Ha, Injong Rhee, and Lisong Xu. CUBIC : a New TCP-Friendly High-Speed TCP Variant. *SIGOPS Oper. Syst. Rev.*, 2008.

[6] Haiqing Jiang, Yaogong Wang, Kyunghan Lee, and Injong Rhee. Tackling Bufferbloat in 3G/4G Networks. IMC'12, 2012.

[7] Luigi A. Grieco and Saverio Mascolo. Performance Evaluation and Comparison of Westwood+, New Reno, and Vegas TCP Congestion Control. *SIGCOMM CCR*, 34, 2004.