RFC 7414 : A Roadmap for Transmission Control Protocol (TCP) Specification Documents

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 3 février 2015

Date de publication du RFC : Février 2015

https://www.bortzmeyer.org/7414.html

C'est un RFC récapitulatif. Il ne normalise rien de nouveau mais dresse une liste commentée des RFC dont la connaissance est indispensable, ou simplement utile, au programmeur qui met en œuvre TCP.

Depuis sa normalisation, il y a plus de trente ans (dans le RFC 793 [1]), TCP a complètement remplacé NCP. TCP est un des grands succès de l'Internet : quasiment toutes les applications Internet s'appuient sur ce protocole.

Mais le RFC normatif, le RFC 793, quoique toujours valable à l'époque, était bien vieux car beaucoup de choses avaient été ajoutées ou retirées à TCP depuis. (Mais, depuis la publication de ce RFC 7414, la norme TCP a été enfin remplacée, par le RFC 9293.) Comme pour beaucoup d'autres protocoles Internet (par exemple le DNS), TCP met donc le programmeur en face d'une rude tâche : avant de commencer à coder, il doit d'abord dresser la liste de tous les RFC dont il aura besoin. Et il faudra qu'il pense à chaque fois à regarder si des errata <http://www.rfc-editor.org/errata.php> ont été publiés. C'est cette tâche d'établissement de liste que lui épargne notre RFC en dressant cette liste, et en la rangeant en trois sections, ce qui est impératif pour un TCP moderne (section 2 de notre RFC), ce qui est fortement souhaité (section 3), et ce qu'on peut ajouter si on veut (section 4). Ce « méta-RFC » a donc une bibliographie particulièrement longue, comportant 135 autres RFC. Le groupe de travail avait discuté de l'utilisation d'une autre méthode qu'un RFC, par exemple un Wiki, qui pourrait être plus facilement maintenu à jour, mais sans arriver à se décider pour une autre solution. (Notre nouveau RFC succède au RFC 4614, actualisant ses recommandations.)

Par exemple, le document original sur TCP ne contient rien sur le contrôle de congestion, qui ne sera décrit que dans le RFC 2001. Ce RFC 2001 (ou plus exactement son successeur, le RFC 5681) fait

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc793.txt

désormais partie de ceux qu'il faut lire. Notez que cette importance n'est pas forcément liée au statut officiel du RFC : le RFC 5681 n'est que projet de norme, alors qu'il est considéré essentiel. On trouve aussi dans cette section le RFC 6298, sur le calcul du délai avant retransmission, ou le RFC 6691 sur le calcul de la MSS.

Dans la section de ce qui est fortement recommandé (section 3), on trouve par exemple le RFC 7323 qui décrit plusieurs extensions nécessaires pour tirer des performances élevées, ou bien le RFC 3168 qui normalise ECN, ou encore le RFC 6582, sur l'algorithme NewReno.

Cette section compte également des RFC décrivant l'abandon d'options ou d'extensions inutiles, voire néfastes. C'est le cas du RFC 6633 qui supprime le mécanisme de répression de l'émetteur.

La sécurité ayant bien plus d'importance aujourd'hui, d'autres RFC décrivent comment se défendre contre certaines vulnérabilités par exemple la lecture du RFC 5961 va apprendre aux programmeurs de TCP comment résister aux attaques en aveugle, et celle du RFC 6528 est indispensable pour empêcher un attaquant d'insérer des paquets dans une session TCP existante.

D'autres extensions sont moins consensuelles et restent plutôt expérimentales à ce jour (section 4) comme l'algorithme Eifel du RFC 3522. Certaines de ces extensions non consensuelles sont encore récentes et s'imposeront peut-être, comme l'extension de la fenêtre initiale (RFC 6928) ou comme l'algorithme de réduction proportionnelle (RFC 6937).

Enfin certaines extensions ont été abandonnées, l'expérience ayant montré leur inutilité ou leur nocivité (section 6 du RFC). C'est ainsi que la proposition du RFC 1146 de tenter de nouveaux moyens de calcul de la somme de contrôle n'a pas pris.

Le protocole T/TCP, normalisé dans le RFC 1644, aurait permis de diminuer nettement la durée des connexions courtes, celles où on échange peu de données (beaucoup de connexions HTTP sont dans ce cas). Promu par des experts comme Stevens <http://www.kohala.com/start/ttcp.html>, implémenté dans des systèmes comme FreeBSD (option MSG_EOF de `sendto`), il a été remisé au grenier après que des analyses plus poussées aient montré ses failles de sécurité (il facilite l'utilisation de TCP avec usurpation d'adresses IP). Même sort pour le plus récent RFC 6013 qui décrivait un "*TCP Cookie Transaction*" mais qui n'a finalement pas suscité beaucoup d'intérêt. En revanche, TCP Fast Open (RFC 7413) est actuellement la méthode à la mode pour diminuer la latence.

Les RFC de ces extensions abandonnées ont été reclassifiés comme « intérêt historique seulement » dans le RFC 6247.

Notre RFC décrit ensuite les RFC d'architecture ou de concepts, puis les RFC qui s'appliquent à certains environnements difficiles comme les liaisons satellite qui font l'objet des RFC 2757 et RFC 2760 ou les liaisons fortement asymétriques, comme le sont les lignes ADSL (traitées dans le RFC 3449).

De nombreux autres cas sont ensuite traitées dans notre RFC. Notre implémenteur n'a pas fini de tout lire !

La section 8 couvre enfin un cas délicat : les extensions à TCP qui, bien que largement utilisées, n'ont jamais fait l'objet d'un RFC ni même, souvent, d'une description formelle. C'est par exemple le cas de la prédiction d'en-tête, une méthode développée par Van Jacobson et Mike Karels à la fin des années 1980 pour accélérer le traitement des paquets TCP en essayant de prédire ce qu'allaient contenir leurs en-têtes. On programme un chemin rapide pour les paquets qui sont conformes aux prévisions et un

chemin plus lent pour les paquets (minoritaires) qui surprennent. Van Vacobson avait décrit cette astuce dans un célèbre message de 1988 <ftp://ftp.ee.lbl.gov/email/vanj.88mar10.txt> « *"The idea is that if you're in the middle of a bulk data transfer and have just seen a packet, you know what the next packet is going to look like"* ».

C'était aussi le cas des *"syncookies"*, option indispensable sans laquelle un serveur Internet peut être mis à genoux très vite par une attaque *"SYN flood"* pour laquelle il n'y a même pas besoin de développements spécifiques, l'outil hping suffisant à l'attaquant. Ces petits gâteaux ont finalement été décrits dans le RFC 4987.