RFC 6564 : An uniform format for IPv6 extension headers

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 10 avril 2012

Date de publication du RFC : Avril 2012

https://www.bortzmeyer.org/6564.html

IPv6 permet l'insertion, entre l'en-tête de taille fixe du paquet, et les données de la couche 4, d'un ou plusieurs **en-têtes d'extensions**, qui modifient le comportement du paquet. L'analyse de ces en-têtes d'extension par un programme a toujours été difficile car ils n'ont pas un format cohérent. Ce RFC résout le problème en décrétant que, au moins pour les **futurs** en-têtes, ils devront suivre un format identique, permettant de les sauter sans même en comprendre le contenu.

Ces en-têtes d'extension sont normalisés dans la section 4 du RFC 2460[1]. La plupart suivent le schéma « un octet pour indiquer le type de l'en-tête suivant, un octet pour indiquer la longueur de cet en-tête, puis les données » mais pas tous (notamment, l'en-tête de fragmentation a une syntaxe différente). Si on veut écrire un programme d'analyse des paquets IPv6 <https://www.bortzmeyer.org/analyse-pcap-ipv6.html>, on doit donc connaître tous les types d'en-tête (il n'y en a, après tout, pas beaucoup) et traiter chacun spécialement. Mais cela ne résout pas le problème des **futurs** en-têtes : si un nouvel en-tête est créé (comme cela avait été le cas dans le RFC 5533), et qu'il commence à apparaître sur le réseau, les programmes d'analyse vont souffrir.

Outre les programmes d'analyse, on peut citer d'autres logiciels qui souhaitent regarder à l'intérieur des paquets, y compris avant que le paquet n'atteigne sa destination (section 1 du RFC). Par exemple les pare-feux veulent accéder à la couche transport, et ils doivent le faire à la vitesse du réseau physique (si on a un Ethernet 1 Gb/s, on veut que les paquets soient analysés au même rythme), ce qui implique le recours à des circuits électroniques relativement simples (les ASIC) qui n'ont pas la possibilité de faire tourner du code arbitrairement complexe. Or, aujourd'hui, tout nouvel en-tête IPv6 est à peu près sûr d'invalider ces pare-feux.

Ne peut-on pas s'en tirer en disant qu'il vaut mieux éviter de définir de nouveaux en-têtes d'extension ? C'est l'option que rappelle la section 3, qui note que l'en-tête "*Destination Options*" (section 4.6

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc2460.txt

du RFC 2460) est déjà très général : il permet d'inclure plusieurs options différentes, chacune identifiée par son type <https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xml#ipv6-parameters-2> et encodée sous forme TLV. Le RFC recommande donc que les concepteurs de nouvelles idées pour IPv6 réutilisent l'en-tête "*Destination Options*" autant que possible. Au cas où un concepteur de protocole décide de créer un nouvel en-tête, il **doit** désormais expliquer en détail pourquoi l'en-tête "*Destination Options*" ne convient pas à son but.

Notez que l'en-tête "*Destination Options*", comme son nom l'indique, ne doit normalement être traité que par la machine de destination (les routeurs, notamment, n'ont pas besoin de le gérer ou de le connaître, sauf à des fins de statistique). Si on veut transporter des options qui sont analysées par les routeurs, on a l'en-tête "*Hop-by-hop Options*". En pratique, il n'est pas du tout évident que ce dernier fonctionne : comme chaque routeur du trajet doit en tenir compte, il ouvre une voie d'attaque possible et il semble que bien des routeurs l'ignorent complètement, voire jettent sans autre forme de procès les paquets qui le contiennent. Quoi qu'il en soit, notre RFC 6564 précise qu'il ne faut **pas** créer de nouveaux en-têtes ayant cette sémantique « examen par chaque routeur ». Non seulement la création d'en-têtes est déconseillée, mais elle ne doit se faire qu'avec la sémantique « examen par la destination seulement ».

Bien, maintenant qu'on a sérieusement refroidi les concepteurs de protocole qui auraient pensé à créer un nouvel en-tête, la section 4 fixe les règles à suivre au cas où on ne serait pas encore découragé. Les éventuels nouveaux en-têtes devront désormais tous suivre le modèle de la plupart des en-têtes existants et commencer par un octet indiquant l'en-tête suivant (typiquement, cela sera les données de la couche Transport, les valeurs possibles sont dans un registre <https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml>), puis un octet indiquant la longueur de l'en-tête. Cela devrait rendre un peu plus facile l'analyse des futurs en-têtes, s'il y en a de créés.

Rappelez-vous que cela n'affecte que les **futurs** en-têtes. Pour analyser les existants (notamment l'en-tête "*Fragment*" qui ne suit **pas** ce schéma, cf. section 5), il faut garder le code existant. La section 6 rappele d'ailleurs que le traitement des en-têtes reste un problème pas complètement résolu pour les équipements intermédiaires, par exemple parce qu'il peut y avoir plusieurs en-têtes, chaînés, parce que leur ordre est important, et parce que le traitement des en-têtes peut influer celui de la charge utile. Personnellement, je pense que la première mise en œuvre d'IPv6 qui essaiera d'utiliser vraiment les en-têtes aura de vilaines surprises... Cela sera encore pire avec un nouvel en-tête, qui ne fait pas partie des anciens.

Notez aussi que tout le problème traité par ce RFC a une faible importance pratique : on ne trouve quasiment pas d'en-têtes d'extension dans les paquets IPv6 qui circulent aujourd'hui.

Donc, désormais, une implémentation d'analyse d'IPv6 (comme la mienne <https://www.bortzmeyer.org/analyse-pcap-ipv6.html>, que j'ai eu la flemme de modifier) peut adopter la stratégie suivante :

— Si c'est un en-tête connu, le traiter selon sa spécification.
— Si c'est un en-tête inconnu et que le "*next header*" est un en-tête connu ou bien un protocole de transport connu, sauter l'en-tête inconnu (en utilisant le second octet, qui indique la taille). C'est là la nouveauté de notre RFC 6564.
— Si c'est un en-tête inconnu et que le suivant n'est pas connu non plus, le problème reste ouvert (tenter de suivre la chaîne ? C'est dangereux car on peut se retrouver au milieu d'un protocole de transport inconnu, pour lesquels les règles de notre RFC ne s'appliquent pas) et n'est pas modifié par notre RFC.