RFC 9098 : Operational Implications of IPv6 Packets with Extension Headers

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 20 septembre 2021

Date de publication du RFC : Septembre 2021

https://www.bortzmeyer.org/9098.html

IPv6 souffre de réseaux mal gérés qui se permettent de jeter les paquets ayant des caractéristiques normales, mais qui déplaisent à certains équipements réseau. C'est par exemple le cas des paquets utilisant les en-têtes d'extension. Pourquoi les paquets utilisant ces en-têtes sont-ils souvent jetés ?

Ces en-têtes d'extension (qui n'ont pas d'équivalent en IPv4) sont normalisés dans le RFC 8200 [1], section 4. Ils servent à la fois à des fonctions de base d'IPv6 (comme la fragmentation) et à étendre le protocole si nécessaire. Mais, en pratique, on observe trop souvent que les paquets IPv6 utilisant ces en-têtes d'extension ne passent pas sur certains réseaux. Ce RFC vise à étudier pourquoi, et à expliquer les conséquences. Comme toujours quand on explique un phénomène (la délinquance, par exemple), des gens vont comprendre de travers et croire qu'on justifie le phénomène en question (« expliquer, c[Caractère Unicode non montré [2]]est déjà vouloir un peu excuser », avait stupidement dit un ancien premier ministre français, à propos du terrorisme djihadiste). Le cas est d'autant plus fréquent qu'en français, « comprendre » est ambigu, car il peut désigner l'explication (qui est utile aussi bien aux partisans qu'aux adversaires d'un phénomène) que la justification. Bref, ce RFC explique pourquoi des paquets innocents sont détruits, il ne dit pas qu'il faut continuer à le faire !

La section 4 du RFC explique la différence entre l'en-tête d'un paquet IPv6 et celui d'un paquet de l'ancienne version. Dans les deux versions, le problème est de placer des options dans les en-têtes. Le gros changement est l'utilisation par IPv6 d'un en-tête de taille fixe (40 octets), suivi d'une liste chainée d'en-têtes d'extension permettant d'encoder ces options. Au contraire, IPv4 avait un en-tête de taille variable (entre 20 et 60 octets), avec un champ indiquant sa longueur, champ qu'il fallait lire avant d'accéder à l'en-tête. Le mécanisme d'IPv4 n'était ni pratique, ni rapide, mais celui d'IPv6 n'est pas plus

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc8200.txt
2. Car trop difficile à faire afficher par LaTeX

satisfaisant : il faut lire toute la liste chainée <https://www.bortzmeyer.org/analyse-pcap-ipv6.html> (dont la longueur est quelconque et non limitée) si on veut accéder aux informations au-dessus de la couche 3. Ce n'est pas un problème pour les routeurs, à qui la couche 3 suffit (à part le cas de l'en-tête "*Hop-by-hop options*"), mais c'est ennuyeux pour les autres boitiers situés sur le chemin, comme les pare-feux. Les évolutions récentes d'IPv6 ont aidé, comme l'obligation que tous les en-têtes d'extension soient dans le premier fragment, si le datagramme est fragmenté (RFC 7112), ou comme l'imposition d'un format commun aux futurs en-têtes d'extension (dans le RFC 6564), mais l'analyse de la liste des en-têtes d'extension reste un travail pénible pour les machines.

Le problème est identifié depuis longtemps. Il est analysé dans l'"*Internet-Draft*" draft-taylor-v6ops-fragdr dans draft-wkumari-long-headers, dans draft-kampanakis-6man-ipv6-eh-parsing mais aussi dans des RFC, le RFC 7113, le RFC 8900 et le RFC 9288. Cela a mené à plusieurs changements dans la norme, je vous renvoie aux RFC 5722, RFC 7045, RFC 8021, etc. Plusieurs changements ont été intégrés dans la dernière révision du RFC principal de la norme, le RFC 8200. Ainsi, celui-ci dispense désormais les routeurs de chercher un éventuel en-tête d'extension "*Hop-by-hop options*".

Parallèlement à ces analyses du protocole, diverses études ont porté sur le phénomène des paquets jetés s'ils contenaient des en-têtes d'extension. Cela a été documenté dans « "*Discovering Path MTU black holes on the Internet using RIPE Atlas*" <http://www.nlnetlabs.nl/downloads/publications/pmtu-black-holes-msc-thesis.pdf> », dans « "*IPv6 Extension Headers in the Real World v2.0*" <http://www.iepg.org/2014-07-20-ietf90/iepg-ietf90-ipv6-ehs-in-the-real-world-v2.0.pdf> », également dans « "*Dealing with IPv6 fragmentation in the DNS*" <https://blog.apnic.net/2017/08/22/dealing-ipv6-fragmentation-dns/> » et « "*Measurement of IPv6 Extension Header Support*" <https://www.cmand.org/workshops/202006-v6/slides/2020-06-16-xtn-hdrs.pdf> » ainsi que dans le RFC 7872.

Comment est-ce que les équipements intermédiaires du réseau fonctionnent, et pourquoi est-ce que les en-têtes d'extension IPv6 peuvent parfois les défriser ? La section 6 du RFC rappelle l'architecture de ces machines. Un routeur de cœur de réseau est très différent de l'ordinateur de base, avec son processeur généraliste et sa mémoire de grande taille, ce qui le rend lent, mais souple. Au contraire, la transmission des paquets par le routeur est en général faite par du matériel spécialisé, ASIC ou NPU. (Vous pouvez consulter des explications dans l'exposé fait à l'IETF « "*Modern Router Architecture for Protocol Designers*" <http://www.iepg.org/2015-11-01-ietf94/IEPG-RouterArchitecture-jgs.pdf> » et dans l'article « "*Modern router architecture and IPv6*" <https://blog.apnic.net/2020/06/04/modern-router> ».) Avant de transmettre un paquet reçu sur l'interface de sortie, le routeur doit trouver quelle interface utiliser. Une méthode courante est de prendre les N premiers octets du paquet et de les traiter dans une TCAM ou une RLDRAM, où se fera la détermination du saut suivant (et donc de l'interface de sortie). Le choix du nombre N est crucial : plus il est petit, plus le routeur pourra traiter de paquets, mais moins il aura d'information sur le paquet. (En pratique, on observe des N allant de 192 à 384 octets.) Du fait qu'un routeur se limite normalement à la couche Réseau, et que l'en-tête IPv6 a une taille fixe, envoyer simplement les 40 octets de cet en-tête devrait suffire. Mais lisez plus loin : certains routeurs ou autres équipements intermédiaires sont configurés pour regarder au-delà, et voudraient des informations de la couche Transport, plus dure à atteindre.

Une solution, pour les routeurs qui ne lisent qu'un nombre limité d'octets avant de prendre une décision, est de lire les en-têtes d'extension un par un, et de réinjecter le reste du paquet dans le dispositif de traitement. Ça se nomme la recirculation, c'est plus lent, mais ça marche quelle que soit la longueur de la chaîne des en-têtes d'extension.

Comme le routeur comprend toujours un processeur généraliste (ne serait-ce que pour faire tourner les fonctions de contrôle comme SSH et les protocoles de routage), une autre solution serait d'envoyer à

ce processeur les paquets « compliqués » (mais lire le RFC 6192 d'abord). L'énorme différence de performance entre le processeur généraliste et les circuits spécialisés fait que ce ne serait pas une solution réaliste sauf pour un trafic très modéré. (Le RFC note que certains routeurs ont trois dispositifs de traitement des paquets, et pas seulement deux comme présenté plus haut : outre le processeur généraliste, qui ne leur sert qu'aux fonctions de contrôle, et des circuits matériels spécialisés, ils ont un dispositif logiciel de transmission des paquets. Plus souple que les circuits spécialisés, il n'est pas forcément plus rapide que le processeur généraliste mais, au moins, il n'interfère pas avec les fonctions de contrôle.)

Bon, on l'a déjà dit, mais cela vaut la peine de le répéter : à part le cas (agaçant) de l'en-tête "*Hop-by-hop options*", un routeur, équipement de couche 3 n'a normalement aucun besoin d'aller regarder tous les en-têtes d'extension (qui sont presque tous pour la machine terminale de destination uniquement). Encore moins de faire du DPI et d'aller chercher des informations dans les couches 4 à 7. Mais la section 7 du RFC explique pourquoi certains routeurs le font quand même (il s'agit d'une description de leurs pratiques, pas une approbation ; beaucoup des raisons données sont mauvaises).

D'abord, la répartition de charge et l'ECMP. Si un routeur peut faire passer un paquet via deux interfaces différentes, laquelle choisir ? Alterner (un coup à gauche, un coup à droite) augmenterait la probabilité d'une arrivée des paquets dans le désordre. On préfère la prédictabilité : tous les paquets d'un même flot doivent suivre le même chemin. Comment déterminer le « flot », notion parfois un peu floue ? En restant dans les couches basses, on peut utiliser uniquement les adresses IP de source et de destination mais elles ne sont pas assez discriminantes. En IPv6, on peut en théorie utiliser l'étiquette de flot (normalisé dans le RFC 6437, ainsi que les RFC 6438 et RFC 7098 pour son utilisation) mais elle n'est pas toujours présente, en partie (encore !) par la faute de "*middleboxes*" qui, ignorant ce concept d'étiquette de flot, jetaient les paquets qui en avaient une. (Voir les articles de I. Cunha, « " *IPv4 vs IPv6 load balancing in Internet routes*" <https://www.cmand.org/workshops/202006-v6/slides/cunha.pdf> » et J. Jaeggli, « " *IPv6 flow label : misuse in hashing*" <https://blog.apnic.net/2018/01/11/ipv6-flow-label-misuse-hashing/> ».) En pratique, les systèmes qui ont besoin d'identifier les flots utilisent plutôt une heuristique : le tuple à cinq élements {protocole de transport, adresse IP source, adresse IP destination, port source, port destination}. Ce n'est pas parfait mais, en pratique, cela marche à peu près. On voit que cela nécessite d'accéder à de l'information de la couche Transport, les ports. (Cette heuristique a d'autres limites : par exemple elle ne marche pas si une partie de la couche Transport est chiffrée, comme c'est le cas avec QUIC <https://www.bortzmeyer.org/quic.html>.)

Autre cas où un boitier intermédiaire ressent le besoin d'accéder aux informations des couches supérieures, les ACL. La plupart des pare-feux permettent d'utiliser comme critère de filtrage les numéros de ports, ce qui va nécessiter d'analyser la chaine des en-têtes d'extension IPv6 pour parvenir à la couche Transport. Entre les pare-feux mal faits qui cherchent les informations de la couche Transport immédiatement après l'en-tête fixe (car le programmeur a tout simplement sauté la partie du RFC qui parlait des en-têtes d'extension) et les difficultés posées par le fait, qu'avant le RFC 6564, les en-têtes d'extension inconnus étaient impossibles à analyser, on voit qu'arriver à la couche Transport n'est pas si facile que ça (le RFC dit qu'il est plus compliqué d'analyser IPv6 qu'IPv4, ce qui est très contestable ; mais ce n'est en effet pas plus facile). Cela peut avoir des conséquences en terme de sécurité, par exemple si un méchant ajoute des en-têtes d'extension dans l'espoir que ses paquets ne soient pas correctement identifiés <http://www.insinuator.net/2014/05/a-novel-way-of-abusing-ipv6-extension-headers-to-e> (cf. RFC 7113).

Autre exemple, le réassemblage des paquets fragmentés nécessite un état (mémoriser les fragments en cours de réassemblage), ce qui peut être fatal à la mémoire du pare-feu, notamment en cas d'attaque. Pour les coûts de ce filtrage, voir l'article d'E. Zack « "*Firewall Security Assessment and Benchmarking IPv6 Firewall Load Tests*" <https://www.ipv6hackers.org/files/meetings/ipv6-hackers-1/zack-ipv6hackers1.pdf> ». Cette difficulté peut encourager les pare-feux à jeter les paquets ayant des en-têtes d'extension,

surtout si un administrateur réseaux ignorant dit bien fort « ces en-têtes d'extension ne servent à rien de toute façon » (RFC 7872).

Un problème proche est celui de la protection contre les attaques par déni de service. Filtrer avec les adresses IP peut être fait rès efficacement (cf. RFC 5635) mais ne marche pas, par exemple si l'adresse IP source est usurpée. Une des méthodes utilisées alors est de repérer un motif récurrent dans les paquets envoyés par l'attaquant, et de filtrer sur ce motif (vous pouvez lire ici un exemple avec le DNS <https://www.bortzmeyer.org/dns-netfilter-u32.html>). Là encore, des informations au-delà de la couche Réseau sont nécessaires. Toujours en sécurité, les NIDS ont aussi besoin de plus d'informations que ce que fournit la couche Réseau.

Et, bien sûr, comme toute complexité, les en-têtes d'extension augmentent les risques de sécurité, d'autant plus qu'il y a un cercle vicieux : comme ils sont peu utilisés, le code qui les traite est peu testé, donc il y a des bogues <https://www.freebsd.org/security/advisories/FreeBSD-SA-20:24.ipv6.asc>, donc ils sont peu utilisés, etc.

Donc, un certain nombre de machines intermédiaires, qui assurent des fonctions allant au-delà de ce que fait un « simple » routeur, ont des problèmes avec les en-têtes d'extension IPv6 et choisissent souvent la solution de facilité, qui est de les jeter systématiquement.