RFC 7872 : Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 5 juillet 2016

Date de publication du RFC : Juin 2016

http://www.bortzmeyer.org/7872.html

Normalement, l'Internet est un monde idéal où la machine d'Alice peut envoyer à celle de Bob les paquets qu'elle veut, Bob les recevra intacts (s'il le veut). Dans la réalité, pas mal de machines intermédiaires ne respectent pas ce principe de bout en bout. Il est fréquent que les paquets « inhabituels » soient jetés en route. Cela peut être le résultat d'une politique délibérée (pare-feu appliquant le principe « dans le doute, on jette ») ou bien, souvent, de l'incompétence des programmeurs qui n'ont pas lu les RFC et ne connaissent pas telle ou telle option. Par exemple, IPv6 permet d'ajouter au paquet, entre l'en-tête proprement dit et la charge utile du paquet, un ou plusieurs **en-têtes d'extension**. C'est rare en pratique. Est-ce que les programmeurs des "*middleboxes*" ont fait attention à cette possibilité ? Les paquets ayant ces en-têtes d'extension ont-ils de bonne chances d'arriver au but ? Ce nouveau RFC est un compte-rendu de mesures effectuées dans l'Internet pour essayer de quantifier l'ampleur exacte du problème.

Un point important du travail effectué par les auteurs du RFC est qu'ils ne sont pas contentés de chercher **si** les paquets étaient jetés mais également **où** ils étaient jetés. La différence est politiquement cruciale. Si le paquet IPv6 portant un en-tête d'extension est jeté par la machine de Bob, car Bob n'aime pas ces en-têtes et il a configuré son Netfilter pour les envoyer vers `DROP`, pas de problème : Bob est libre d'accepter ou de refuser ce qu'il veut. Si, par contre, le paquet a été jeté par le FAI d'Alice ou de Bob, ou, encore pire, par un opérateur de transit entre les deux FAI, c'est grave, c'est une violation de la neutralité du réseau, violation qui empêche deux adultes consentants de s'envoyer les paquets qu'ils veulent. Le mécanisme de mesure cherche donc à trouver dans quel AS le paquet a disparu.

Les mesures ont été faites en août 2014 et juin 2015. Le RFC détaille suffisamment la façon dont elles ont été faites (annexe A) pour qu'une autre équipe puisse recommencer les mesures à une date ultérieure, pour voir si la situation s'améliore.

Les en-têtes d'extension IPv6 sont décrits dans le RFC 2460[1], section 4. Ils sont de deux plusieurs sortes, entre autre :

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

— L'en-tête de fragmentation, qui doit pouvoir passer si on veut que les paquets IP fragmentés arrivents (les principaux émetteurs de paquets IPv6 fragmentés sont sans doute les serveurs DNS, notamment avec DNSSEC, car UDP ne négocie pas la MSS).
— L'en-tête « Options pour la destination » (*"Destination Options"*) qui, comme son nom l'indique, ne doit pas être examiné par les routeurs sur le chemin. Il contient des informations pour la machine terminale de destination.
— L'en-tête « Options pour chaque intermédiaire » (*"Hop-by-Hop Options"*) qui, comme son nom l'indique, doit être examiné par les routeurs sur le trajet. il contient des informations pour eux.

Outre la fragmentation, un exemple d'utilisation de ces en-têtes est le protocole CONEX (RFC 7837). Si vous voulez ajouter des options pour la destination dans un paquet IPv6, vous pouvez regarder mon tutoriel <http://www.bortzmeyer.org/destination-options-ipv6.html>.

Quelles sont les chances de survie des paquets portant ces en-têtes d'extension dans le sauvage Internet ? Des études comme « *"Discovering Path MTU black holes on the Internet using RIPE Atlas"* <http://www.nlnetlabs.nl/downloads/publications/pmtu-black-holes-msc-thesis.pdf> », « *"Fragmentation and Extension header Support in the IPv6 Internet"* <http://www.iepg.org/2013-11-ietf88/fgont-iepg-ietf88-ipv6-frag-and-eh.pdf> » ou bien « *"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-0.pdf> » s'étaient déjà penchées sur la question. Les mesures de ce RFC détaillent et complètent ces résultats préliminaires, notamment sur la différenciation entre perte de paquet près de la destination, et perte du paquet en transit. La conclusion est, qu'hélas, les paquets IPv6 portant des en-têtes d'extension font effectivement l'objet d'une discrimination négative (« entêtophobie » ?).

Donc, qu'est-ce que ça donne dans l'Internet réel (section 2) ? Deux listes de serveurs IPv6 ont été utilisées, celle du *"World IPv6 Launch Day"* <http://www.worldipv6launch.org/> et le premier million d'Alexa <http://www.alexa.com/>. De ces listes de domaines ont été extraits les adresses IP des sites Web (`dig AAAA LE-DOMAINE` et, oui, je sais que c'est un sérieux raccourci de supposer que tout domaine a un serveur Web à l'apex), des relais de messagerie (`dig MX LE-DOMAINE` puis obtenir l'adresse IPv6 des serveurs), et des serveurs de noms (`dig NS LE-DOMAINE`). Les adresses absurdes (`::1`...) ont été retirées. Ensuite, pour chaque adresse ainsi obtenue, trois types de paquets ont été envoyés :

— Un avec en-tête d'extension *"Destination Options"*,
— Un avec en-tête d'extension *"Hop-by-hop Options"*,
— Un fragmenté en deux fragments d'à peu près 512 octets.

Les paquets étaient tous du TCP à destination d'un port correspondant au service (25 pour les serveurs de messagerie, par exemple).

Un point très important et original de cette expérience était la recherche d'information sur **où** se produisait l'éventuelle perte de paquets (dans quel AS). L'hypothèse de base est qu'une élimination du paquet dans l'AS de destination **peut** être acceptable, parce qu'elle résulte d'une décision consciente du destinataire. En tout cas, il sera plus facile de changer la politique de l'AS de destination. En revanche, une élimination dans un AS intermédiaire est inacceptable : elle indique filtrage ou erreur de configuration dans un réseau qui devrait être neutre. Notez que l'hypothèse n'est pas parfaite : si un particulier héberge un serveur chez lui et que son FAI filtre les paquets IPv6 avec en-tête d'extension, c'est quand même inacceptable, et c'est difficile pour le particulier de le faire corriger.

Comment identifier l'AS qui jette le paquet ? L'idée est de faire l'équivalent d'un traceroute (en fait, deux, un avec l'en-tête de destination et un sans, pour comparer : cf. annexe B.1) et de voir quel est le dernier routeur qui a répondu. C'est loin d'être idéal. Des routeurs ne répondent pas, pour d'autres, il est difficile d'évaluer à quel AS ils appartiennent (dans un lien de *"peering"* entre deux acteurs A et B, les adresses IP utilisées ont pu être fournies par A ou bien par B ; cf. annexe B.2). Et l'absence d'un routeur dans le résultat de traceroute ne prouve pas que le routeur n'a pas transmis le paquet juste avant. La mesure indique donc deux cas, l'optimiste, où on suppose, en l'absence de preuve opposée, que les paquets manquant ont été jetés par l'AS de destination et le pessimiste où on fait la supposition inverse.

Les résultats complets des mesures figurent dans le RFC. Je résume ici une partie. Pour les serveurs Web de la liste du *"World IPv6 Launch Day"* :

— 12 % des serveurs n'ont pas reçu le paquet envoyé avec un en-tête "*Destination Options*". Entre 18 % (optimiste) et 21 % (pessimiste) de ces pertes se sont produites avant l'AS de destination.
— 41 % des serveurs n'ont pas reçu le paquet envoyé avec un en-tête "*Hop-by-hop Options*". Entre 31 % (optimiste) et 40 % (pessimiste) de ces pertes se sont produites avant l'AS de destination.
— 31 % des serveurs n'ont pas reçu le paquet fragmenté. Entre 5 % (optimiste) et 7 % (pessimiste) de ces pertes se sont produites avant l'AS de destination.

Les résultats ne sont guère différents pour les autres services (comme SMTP), indiquant que le problème, quand il est présent, est bien dans la couche 3. Idem avec le jeu de données Alexa : peu de différences.

On voit que les paquets utilisant l'en-tête d'extension « Options pour chaque intermédiaire » sont ceux qui s'en tirent le plus mal (c'est un peu logique, cet en-tête étant censé être examiné par tous les routeurs intermédiaires). Les fragments passent assez mal également. Enfin, on note qu'un pourcentage significatif des paquets sont jetés par un AS de transit, qui viole ainsi massivement le principe de bout en bout. Triste état de l'Internet actuel, pourri de "*middleboxes*" qui se permettent tout.

Si vous voulez refaire l'expérience vous-même, pour contrôler les résultats, ou bien pour mesurer l'évolution dans quelque temps, l'annexe A vous donne les éléments nécessaires. Vous aurez besoin du toolkit de SI6 <http://www.si6networks.com/tools/ipv6toolkit>. Les données d'Alexa sont disponibles en ligne <http://s3.amazonaws.com/alexa-static/top-1m.csv.zip>. L'outil `script6` dans le "*toolkit*" SI6 permet d'obtenir les adresses IP nécessaires. Par exemple :

```
% cat top-1m.txt | script6 get-mx | script6 get-aaaa
```

Et on obtient ainsi les adresses IP des relais de messagerie. Pour supprimer les adresses incorrectes (par exemple `::1`), on utilise l'outil `addr6` du "*toolkit*" :

```
%  cat top-1m-mail-aaaa.txt | addr6 -i -q -B multicast -B unspec -k global
```

Il n'y a plus maintenant qu'à envoyer les paquets portant les en-têtes d'extension, ou fragmentés. Ici, les serveurs de messagerie avec l'en-tête "*Destination Options*" (« do8 ») :

```
% cat top-1m-mail-aaaa-unique.txt | script6 trace6 do8 tcp 25
```

L'annexe B de notre RFC contient quelques avertissements méthodologiques.

Un traceroute ordinaire ne suffit pas à faire les mesures décrites plus haut pour identifier le routeur responsable d'une perte de paquet (le traceroute ordinaire ne permet pas d'ajouter l'en-tête d'extension aux paquets utilisés). Il faut utiliser un outil spécial comme le `path6` du "*toolkit*" SI6. Encore plus simple, dans le même paquetage, l'outil `blackhole6`. Voyons d'abord `path6` pour comprendre les détails :

```
% sudo path6  -d yeti.ipv6.ernet.in
Tracing path to yeti.ipv6.ernet.in (2001:e30:1c1e:1::333)...

  1 (2001:4b98:dc0:41::250)   1.1 ms   0.5 ms   0.4 ms
  2 (2001:4b98:1f::c3d3:249)   1.2 ms   3.5 ms   3.4 ms
  3 (2001:7f8:54::173)   1.1 ms   0.8 ms   0.6 ms
  4 (2001:1a00:1:cafe::e)  141.3 ms  140.7 ms  140.6 ms
  5 (2001:1a00:1:cafe::145)  118.8 ms  118.8 ms  119.5 ms
  6 (2001:1a00:1:cafe::141)  140.7 ms  137.2 ms  137.3 ms
  7 (2001:1a00:1:cafe::10b)  144.0 ms  144.3 ms  144.1 ms
  8 (2001:1a00:1:cafe::212)  140.5 ms  140.5 ms  140.5 ms
  9 (2001:1a00:1:cafe::207)  137.0 ms  137.0 ms  136.9 ms
 10 (2001:1a00:acca:101f::2)  136.5 ms  136.4 ms  136.5 ms
 11 (2001:4528:ffff:ff04::1)  152.2 ms  152.1 ms  152.2 ms
 12 (2001:4528:fff:c48::1)  163.6 ms  163.7 ms  163.6 ms
 13 (2001:e30:1c1e::1)  162.6 ms  162.3 ms  162.2 ms
 14 (2001:e30:1c1e:1::333)  174.5 ms  175.1 ms  175.2 ms
```

OK, on arrive à joindre la machine en Inde. Maintenant, ajoutons l'en-tête "*Hop-by-Hop Options*" :

```
% sudo path6 --hbh-opt-hdr 8 -d yeti.ipv6.ernet.in
Tracing path to yeti.ipv6.ernet.in (2001:e30:1c1e:1::333)...

  1 (2001:4b98:dc0:41::250)  20.9 ms  20.0 ms  19.0 ms
  2 (2001:4b98:1f::c3d3:249)  20.8 ms  20.2 ms  18.5 ms
  3 (2001:4b98:1f::c3c4:3)  20.7 ms  20.4 ms  18.5 ms
  4 (2001:1a00:1:cafe::e)  154.3 ms  14.4 ms  152.1 ms
  5 (2001:1a00:1:cafe::e)  151.0 ms  129.6 ms  148.9 ms
  6 (2001:1a00:1:cafe::141)  151.2 ms  126.2 ms  148.8 ms
  7 (2001:1a00:1:cafe::141)  156.5 ms  156.8 ms  158.6 ms
  8 (2001:1a00:1:cafe::212)  153.6 ms  191.2 ms  151.4 ms
  9 (2001:1a00:1:cafe::212)  155.3 ms  151.1 ms  157.7 ms
 10 (2001:1a00:1:cafe::207)  *  155.619995 ms *
 11 ()   *  *  *
 12 ()   *  *  *
 13 ()   *  *  *
 ...
```

Aïe, ça n'arrive plus. Le routeur situé après `2001:1a00:1:cafe::207` a jeté le paquet. En comparant les deux `path6`, on peut voir que le coupable est `2001:1a00:acca:101f::2` (Flag, désormais une marque de Reliance). L'outil `blackhole6` fait tout ce travail automatiquement (EH = "*Extension Header*", HBH 8 = "*Hop-by-Hop, size 8*") :

```
% sudo blackhole6 yeti.ipv6.ernet.in hbh8
SI6 Networks IPv6 Toolkit v2.0 (Guille)
blackhole6: A tool to find IPv6 blackholes
Tracing yeti.ipv6.ernet.in (2001:e30:1c1e:1::333)...

Dst. IPv6 address: 2001:e30:1c1e:1::333 (AS2697 - ERX-ERNET-AS Education and Research Network, IN)
Last node (no EHs): 2001:e30:1c1e:1::333 (AS2697 - ERX-ERNET-AS Education and Research Network, IN) (14 hop
Last node (HBH 8): 2001:1a00:1:cafe::207 (AS15412 - FLAG-AS Flag Telecom Global Internet AS, GB) (10 hop(s))
Dropping node: 2001:1a00:acca:101f::2 (AS15412 - FLAG-AS Flag Telecom Global Internet AS, GB || AS Unknown -
```

Sur ce sujet et ces tests, on peut aussi regarder l'exposé de Mehdi Kouhen <`http://pirl.tech/pdf/Mehdi_Kouhen_SymposiumV2.pdf`> au symposium Polytechnique/Cisco, ou bien celui d'Eric Vyncke <`https://www.protosec.org/talks.html#evynckeehs`> à Protosec à Buenos Aires.
