Fragmentation IPv6 : se résigner à couper à 1280 octets ?

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 12 novembre 2010

https://www.bortzmeyer.org/fragmentation-ip-1280.html

Les réunions IETF sont toujours l'occasion de discussions intéressantes à table. À l'IETF 79 <http://www.ietf.org/meeting/79/index.html> à Pékin, une vive discussion a ainsi fait irruption au dessus du canard laqué de midi : ne faudrait-il pas changer les algorithmes utilisés pour faire de la fragmentation en IPv6 ? Rémi Després (auteur du RFC 5569 [1]) suggérait en effet de ne pas envoyer de paquets de plus de 1280 octets lorsqu'on sort du réseau local.

Un petit détour d'abord, avant de discuter de cette proposition. Comment fonctionne la fragmentation dans IP ? Elle est très différente en IPv4 et en IPv6. En IPv4, l'émetteur d'un paquet (le RFC sur IPv4, le RFC 791 parle en général de "*datagram*" et celui sur IPv6, le RFC 2460, de "*packet*", je vais juste dire paquet en considérant que c'est un synonyme de datagramme) suit la MTU du réseau local (typiquement 1500 octets, la MTU d'Ethernet). Si, plus tard sur le chemin, un lien a une MTU plus faible (par exemple parce qu'on entre dans un tunnel), le routeur qui émet sur ce lien **fragmente** le paquet (RFC 791, section 2.3) . Comme cette fragmentation ralentit les routeurs (et que le réassemblage ultérieur ralentit les machines de destination), une optimisation existe, la PMTUD ("*Path MTU Discovery*", RFC 1191), qui permet de découvrir la MTU du chemin complet (c'est la MTU du lien ayant la plus petite) et donc d'envoyer uniquement des paquets d'une taille telle qu'ils passeront. Cette méthode est peu fiable en pratique (RFC 2923) en raison notamment de l'incompétence d'administrateurs réseaux qui bloquent tout ICMP. On pourrait certes résoudre le problème en massacrant sauvagement tous les incompétents mais cela prendrait du temps. Et, pour IPv4, ce n'est pas forcément indispensable puisque, si l'émetteur ne peut pas fragmenter, les routeurs intermédiaires le feront.

Mais IPv6 est très différent : cette fois, les routeurs intermédiaires n'ont plus le droit de fragmenter, seule la machine émettrice le peut. Elle doit donc faire quelque chose du genre (en pseudo-code) :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5569.txt

```
if is_link_local(destination) {
    packet_size := mtu(interface)
}
else { /* Destination distante */
   packet_size := path_mtu_discovery(destination)
}
/* Ensuite, on découpe les données en paquets de taille packet_size. */
```

(Pour un exemple réel, voir `net/ipv6/ip6_output.c` dans Linux ou `sys/netinet6/ip6_output.c` sur FreeBSD.) L'algorithme de PMTUD (spécifié, pour IPv6, dans le RFC 1981) devient alors indispensable.

Or, il n'est pas plus fiable en IPv6 qu'en IPv4, alors même que le pourcentage plus élevé de tunnels, ayant une MTU ¡ 1500 octets, le rend plus nécessaire. Résultat, tous les gens qui essaient de faire de l'IPv6 pour de bon (et pas seulement d'en parler dans les colloques) souffrent de ce genre de problèmes. Par exemple, un pare-feu configuré avec les pieds bloque tous les paquets ICMP (une énorme erreur de configuration, mais très fréquente), alors qu'un tunnel, par exemple 6rd, se trouve sur le chemin, avec une MTU plus faible que 1500 octets. Résultat, le message ICMP ≪ Attention, ce paquet est trop gros, il ne passera pas ≫ n'est pas reçu par l'émetteur. La PMTUD déduit donc à tort que la MTU du chemin est de 1500 octets, IP envoie des paquets de cette taille... et ils ne sont jamais reçus. Pour un exemple concret, voir, par exemple, mon exposé à l'OARC <`https://www.dns-oarc.net/files/workshop-201010/expose-octobre-2010.pdf`> sur les conséquences de la signature DNSSEC de `.fr`, qui a augmenté la taille des paquets au delà de 1500.

Il existe des solutions ponctuelles (comme le RFC 4821 qui est spécifique à TCP et n'aurait donc pas aidé pour des problèmes UDP comme ceux que je cite plus haut) mais pas de solution générale (à part l'extermination totale des administrateurs réseaux qui bloquent l'ICMP).

Donc, que proposait Rémi Després ? Avant de résumer sa proposition, je précise que c'est **mon** résumé et qu'il n'est pas responsable de mes éventuelles erreurs ou incompréhensions. Donc, son idée, si j'ai bien compris, est d'admettre que la découverte de la MTU du chemin complet ne fonctionne pas réellement et d'exploiter le fait qu'IPv6 impose une MTU minimale qui est assez raisonnable : 1280 octets (RFC 2460, section 5, elle n'était que de 68 octets en IPv4). On peut donc renoncer à la PMTUD et remplacer l'algorithme présenté plus haut par :

```
if is_link_local(destination) {
    packet_size := mtu(interface)
}
else { /* Destination distante */
   packet_size := 1280
}
/* Ensuite, on découpe les données en paquets de taille packet_size. */
```

Ce n'est peut-être pas très glorieux mais au moins cela marchera dans tous les cas. (Une version encore moins glorieuse serait obtenue en abaissant la MTU du lien à 1280, par exemple avec `ifconfig eth0 mtu 1280`. Dans ce cas, même les paquets pour le réseau local seraient plus petits.) Les programmeurs FreeBSD noteront que ce système dispose d'une option de `setsockopt()`, `IPV6_USE_MIN_MTU`, qui met exactement en œuvre cet algorithme. Voir `ip6(4)`. Certains protocoles IPv6 comme Teredo ont déjà une telle règle (cf. RFC 4380, section 5.1.2.

Quelles seraient les conséquences pratiques d'un tel algorithme ? D'une part une augmentation de la taille relative des en-têtes : l'en-tête IPv6 étant de taille fixe, plus le paquet est petit, plus la part des

en-têtes (la "*header tax*") est élevée (elle passerait de 2,7 % à 3,1 % mais ce calcul ne s'applique qu'aux paquets de la taille maximale ; dans une session TCP, la moitié des paquets sont des accusés de réception, de bien plus petite taille). D'autre part, le coût de traitement d'un paquet dans les équipements réseau comme les routeurs et les commutateurs est relativement indépendant de leur taille. En réduisant la taille des paquets, on augmente leur nombre (de 17 %) et donc la charge des équipements réseau. Alors qu'il faudrait plutôt augmenter la taille des paquets <http://staff.psc.edu/mathis/MTU/> au delà de 1500 octets pour les réseaux à haute performance d'aujourd'hui, se résigner à une taille réduite serait dommage. Mais peut-être n'aura-t-on pas le choix...

---

https://www.bortzmeyer.org/fragmentation-ip-1280.html