

RFC 1191 : Path MTU discovery

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 20 Mai 2007

Date de publication du RFC : Novembre 1990

<http://www.bortzmeyer.org/1191.html>

Ce RFC fut le premier à proposer une méthode pour déterminer la MTU disponible entre deux nœuds de l'Internet.

Une telle détermination sert à beaucoup de choses mais notamment à éviter la fragmentation en envoyant des paquets qui pourront arriver à destination « tels quels ». La fragmentation réduisant le débit, il est souhaitable de l'éviter dès le début. On peut lire à ce sujet "*Raising the Internet MTU*" <<http://www.psc.edu/~mathis/MTU/>> qui décrit en détail pourquoi il serait bon d'augmenter les MTU. (Et, pour un point de vue plus sceptique, l'excellent article de Simon Leinen <<http://www.gossamer-threads.com/lists/nanog/users/96730#96730>>).

En attendant, qu'elle que soit la MTU, il est préférable de la déterminer dès la source des paquets, plutôt que de compter sur les routeurs ultérieurs pour fragmenter (en IPv6, c'est même obligatoire, les routeurs n'ayant pas le droit de fragmenter).

L'algorithme proposé par notre RFC est le suivant : envoyer des paquets avec le bit DF ("*Don't fragment*") mis et voir si on reçoit des paquets ICMP "*Datagram too big*" qui contiennent en général la MTU maximale du lien suivant. Voici, vu par tcpdump, un de ces paquets ICMP, indiquant que le lien suivant ne peut faire passer que 1492 octets d'un coup :

```
11:10:23.474673 IP 172.19.1.1 > 172.19.1.2: icmp 556: 192.134.4.69 unreachable - need to frag (mtu 1492)
```

On recommence avec la nouvelle taille jusqu'à ce que le paquet atteigne sa destination. On apprend ainsi la MTU maximale du lien (c'est la plus petite MTU de tous les liens intermédiaires) et on peut ensuite utiliser des paquets de la taille optimale.

Sur le papier, la méthode est imparable. Mais, en pratique, il est fréquent qu'elle ne marche pas. Beaucoup de sites, en effet, filtrent stupidement tous les paquets ICMP et la machine qui tente de faire de la "*Path MTU discovery*" n'aura jamais de réponse.

Aujourd'hui, on doit donc plutôt utiliser des bricolages comme le "*MSS clamping*" qui consiste à modifier sauvagement la MSS des paquets TCP. Le "*clamping*" est décrit par exemple en <http://www.netbsd.org/Documentation/network/pppoe/#clamping>, pour NetBSD. Avec Linux, on peut utiliser la technique décrite en <http://www.linux.com/howtos/Adv-Routing-HOWTO/lartc.cookbook.mtu-mss.shtml> ou bien l'option `-m` du programme `pppoe`. Par exemple, pour ma connexion ADSL, mon fichier `/etc/ppp/peers/monFAI` contient `pppoe -I eth1 -T 80 -m 1412` pour « clamber » le MSS de TCP à 1412 octets.

Le "*MSS clamping*" ne fonctionne qu'avec TCP. Une autre solution, sur un réseau local qui ne peut pas faire sortir des paquets de 1500 octets (par exemple car il est connecté à Internet avec PPPoE) est de changer la MTU sur toutes les machines du réseau (commande `ifconfig` sur Unix).

Mais une autre approche est possible depuis peu : le RFC 4821¹ décrit un moyen de découvrir la MTU maximale sans dépendre des paquets ICMP. Il reste donc à voir s'il sera largement déployé.

1. Pour voir le RFC de numéro NNN, <http://www.ietf.org/rfc/rfcNNN.txt>, par exemple <http://www.ietf.org/rfc/rfc4821.txt>