Le bonheur des globes oculaires (IPv6 et IPv4)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 1 décembre 2011

http://www.bortzmeyer.org/globes-oculaires-heureux.html

Le déploiement très lent d'IPv6 dans l'Internet fait réfléchir aux causes : pourquoi est-ce que cela ne va pas plus vite ? Pour le cas des sites Web regardés par un public passif (public poétiquement baptisé « globes oculaires » - "*eyeballs*"), une des raisons est que d'être accessible en IPv6 peut rendre l'accès très lent pour les clients qui ont une connexion IPv6 incorrecte. Pourquoi ? Et quels sont les derniers développements qui permettent de traiter ce problème ?

Commençons par définir le problème : soit le célébrissime M. Toutlemonde qui veut regarder un site Web avec du graphique plein partout. M. Toutlemonde a donc des "*eyeballs*", des globes oculaires, qui attendent avec impatience de se délecter d'images qui bougent. Mettons que le site Web visé a une connexion IPv4, une en IPv6 et qu'il publie dans le DNS les deux adresses (A et AAAA). Mettons encore que M. Toutlemonde **a** une connexion IPv6 mais qu'elle est très mauvaise, voire complètement cassée (c'est fréquent avec des techniques comme Teredo, 6to4, ou des tunnels d'amateurs, et c'est pour cela qu'il faut fuir toutes ces techniques).

Le navigateur Web de M. Toutlemonde va tenter une connexion en IPv6 (c'est ce que demande le RFC 6724 [1]), la connexion IPv6 étant inutilisable, ce logiciel va attendre... attendre... attendre une réponse qui ne viendra jamais. Il passera finalement à l'adresse suivante, en IPv4, mais trop tard. M. Toutlemonde, furieux, aura déjà éteint son ordinateur et ouvert un livre en papier de Frédéric Mitterrand à la place. Comment lui éviter ce sort tragique ?

L'algorithme utilisé par les programmeurs « naïfs » n'est pas optimum. Cet algorithme, c'est, en gros, la méthode séquentielle suivante :

```
adresses = getaddrinfo(nom_du_serveur)
pour chaque adresse dans adresses
     connexion(adresse)
     si connexion réussie, sortir de la boucle
```

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6724.txt

L'étape `connexion(adresse)` est bloquante et prend un temps fou si l'adresse en question est injoignable. C'est à cause de cela que l'écrasante majorité des gros sites Web ne publient pas d'adresse IPv6 dans le DNS, de peur de mécontenter les globes oculaires de leurs utilisateurs.

La première idée d'optimisation a donc été de faire les tentatives de connexion en **parallèle** et non plus séquentiellement. Mais cette technique a des inconvénients aussi : elle consomme N fois plus de paquets (pour N adresses disponibles) et elle ouvre plusieurs connexions avec le serveur si celui-ci a plusieurs adresses qui répondent, connexions qu'il faudra fermer ensuite.

D'où la méthode qui est à la mode (illustrée par un célèbre programme de l'ISC <`http://www.isc.org/community/blog/201101/how-to-connect-to-a-multi-homed-server-over-tcp`> ou bien par ce code (en ligne sur `http://www.bortzmeyer.org/files/multi-connect.go`) en Go) : tenter la connexion en IPv6 mais avec un délai de garde très court, et passer aux adresses v4 plus vite, sans attendre le délai de garde normal d'une tentative de connexion TCP. 300 à 500 mili-secondes suffisent : il est rare qu'une connexion réussisse en un temps plus long. Ce système permet-il d'avoir toujours des « "*happy eyeballs*" », des globes oculaires heureux ?

Si vous avez logiciel et connexion IPv6, vous pouvez tester vous-même avec votre navigateur favori, en regardant des sites Web dont la connectivité IPv6 est délibérement en panne, comme <`http://intentionally.broken.dualstack.wdm.sg.ripe.net/`> (admirez son adresse IPv6...) ou <`http://broken.redpill-linpro.com/`>. Si vous y arrivez très vite, vous être un globe oculaire heureux, derrière un navigateur Web qui met en œuvre correctement les dernières techniques. Si vous utilisez un logiciel qui utilise l'ancien algorithme (ici, wget), vous allez souffrir :

```
% time wget http://intentionally.broken.dualstack.wdm.sg.ripe.net/
--2011-12-01 22:30:39--  http://intentionally.broken.dualstack.wdm.sg.ripe.net/
Resolving intentionally.broken.dualstack.wdm.sg.ripe.net... 2001:67c:2e8:dead:dead:dead:dead:dead, 193.0.0.1
Connecting to intentionally.broken.dualstack.wdm.sg.ripe.net|2001:67c:2e8:dead:dead:dead:dead:dead|:80...
[Vingt secondes s'écoulent ici...]
   failed: Connection timed out.
Connecting to intentionally.broken.dualstack.wdm.sg.ripe.net|193.0.0.169|:80... connected.
HTTP request sent, awaiting response... 200 OK
```

Mais un excellent article d'Emile Aben, « "*Hampering Eyeballs - Observations on Two "Happy Eyeballs" Implementations*" <`https://labs.ripe.net/Members/emileaben/hampered-eyeballs`> », montre que le problème est plus compliqué que cela. Il a testé deux implémentations de l'idée ci-dessus, une sur Google Chrome (ticket #81686 <`http://code.google.com/p/chromium/issues/detail?id=81686`>) et l'autre sur deux navigateurs tournant sur Mac OS X (l'algorithme de Mac OS X a été décrit par Apple <`http://lists.apple.com/archives/ipv6-dev/2011/Jul/msg00009.html`>). L'analyse montre que Chrome est parfait et réussit à rendre les globes oculaires heureux dans tous les cas. Mais Mac OS X ne s'y est pas aussi bien pris. Avec le navigateur Firefox, les globes oculaires peuvent être dans certains cas (machine qui a démarré récemment et n'a pas encore accumulé de statistiques de performance) aussi malheureux qu'avant. Avec Safari, le pire est évité mais il n'utilise parfois pas des connexions IPv6 pourtant parfaitement opérationnelles.

Si vous voulez tester vous-même le niveau de bon ou mauvais fonctionnement de serveurs IPv6, je recommande un programme attaché au système de gestion de bogues de Mozilla <`https://bug614526.bugzilla.mozilla.org/attachment.cgi?id=494339`> :

```
% ./test-happiness-eyeballs broken.redpill-linpro.com www.bortzmeyer.org \
             intentionally.broken.dualstack.wdm.sg.ripe.net www.ietf.org
[         0us] begin gai_and_connect(broken.redpill-linpro.com)
[+     1626us] getaddinfo(broken.redpill-linpro.com) done
```

```
[+       34us] dest = 2a02:c0:1002:11::dead (AF_INET6)
[+       13us] about to connect()
[+ 20997111us] connect() fails: Connection timed out
[+       77us] dest = 87.238.47.15 (AF_INET)
[+       14us] about to connect()
[+    59079us] connect() suceeds

[         0us] begin gai_and_connect(www.bortzmeyer.org)
[+      621us] getaddinfo(www.bortzmeyer.org) done
[+       18us] dest = 2001:4b98:dc0:41:216:3eff:fece:1902 (AF_INET6)
[+       11us] about to connect()
[+   156535us] connect() suceeds
[+       52us] dest = 2605:4500:2:245b::bad:dcaf (AF_INET6)
[+       13us] about to connect()
[+   117633us] connect() suceeds
[+       65us] dest = 204.62.14.153 (AF_INET)
[+       14us] about to connect()
[+   111054us] connect() suceeds

[         0us] begin gai_and_connect(intentionally.broken.dualstack.wdm.sg.ripe.net)
[+      558us] getaddinfo(intentionally.broken.dualstack.wdm.sg.ripe.net) done
[+       19us] dest = 2001:67c:2e8:dead:dead:dead:dead:dead (AF_INET6)
[+       10us] about to connect()
[+ 20998608us] connect() fails: Connection timed out
[+       58us] dest = 193.0.0.169 (AF_INET)
[+       11us] about to connect()
[+    38271us] connect() suceeds

[         0us] begin gai_and_connect(www.ietf.org)
[+      635us] getaddinfo(www.ietf.org) done
[+       18us] dest = 2001:1890:123a::1:1e (AF_INET6)
[+       10us] about to connect()
[+   190011us] connect() suceeds
[+       61us] dest = 12.22.58.30 (AF_INET)
[+       14us] about to connect()
[+   199703us] connect() suceeds
```

Vous voyez que les connexions qui réussissent le font en moins de 200 ms, alors que celles qui échouent prennent 20 s.

Depuis la publication de cet article, Christophe Wolfhugel me fait remarquer qu'il n'y a pas que l'établissement de la connexion à prendre en compte pour le bonheur des globes oculaires. En effet, IPv6 a plus souvent des problèmes de MTU qu'IPv4 (les tunnels y sont plus fréquents, et l'ICMP plus souvent bloqué par des administrateurs réseaux maladroits, cf. RFC 2923 et RFC 4459). Dans ce cas, le globe oculaire malheureux risque de voir une connexion qui réussit en IPv6 (les paquets TCP de la poignée de main initiale sont tous de petite taille, bien inférieure à la MTU) mais où on ne pourra pas envoyer de données ensuite (car les paquets de données auront, eux, la taille maximale). Je ne connais pas de site Web délibérement cassé de ce point de vue (pour pouvoir tester) mais, en gros, si votre navigateur affiche qu'il a pu se connecter mais qu'ensuite il attend, cela peut être un problème de MTU (voir mon article à ce sujet <http://www.bortzmeyer.org/mtu-et-mss-sont-dans-un-reseau.html>, en attendant que tout le monde mette en œuvre le RFC 4821).

Un autre article sur le même sujet, très détaillé techniquement avec analyse du comportement des paquets, est « "*Dual Stack Esotropia*" <http://www.ipjforum.org/?p=628> ». La question du bonheur des globes oculaires a depuis fait l'objet du RFC 6555 qui spécifie les algorithmes utiles et du RFC 6556 qui décrit une technique de mesure du niveau de bonheur.
