

RFC 4656 : A One-way Active Measurement Protocol (OWAMP)

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 27 septembre 2006. Dernière mise à jour le 11 septembre 2008

Date de publication du RFC : Septembre 2006

<https://www.bortzmeyer.org/4656.html>

Mesurer les performances d'une connexion à l'Internet est évidemment essentiel pour tous. Ce RFC propose un protocole, OWAMP, pour effectuer des mesures à sens unique (c'est-à-dire ne nécessitant pas que la réponse revienne).

Tout le monde a besoin de savoir si une liaison Internet fonctionne bien : du client qui veut vérifier les promesses de son fournisseur à l'ingénieur réseaux qui essaie de déboguer un problème de performances, avec les utilisateurs qui crient dans son dos "Ça rame!". Traditionnellement, ces mesures de performance étaient effectuées sur des trajets aller-retour ("*round-trip*"). Ainsi, le logiciel ping mesure le temps d'aller-retour (RTT pour "*round-trip time*"). Le RFC 5357¹ normalise un protocole de mesure de RTT, TWAMP. Non seulement cela nécessite que le chemin de retour fonctionne, mais cela interdit de séparer le temps de parcours aller du temps de parcours retour.

Le groupe de travail IETF IPPM ("*IP Performance Metrics*") a donc produit de nombreux RFC spécifiant les métriques à utiliser pour des mesures de performance à sens unique (notamment les RFC 7679 et RFC 7680). En pratique, ces mesures se font en utilisant deux horloges très précises (par exemple GPS) sur les deux machines et en envoyant des paquets depuis la machine sonde vers la machine réceptrice.

Notre nouveau RFC franchit une nouvelle étape en normalisant le protocole de communication entre les deux machines, afin qu'elles puissent se mettre d'accord sur le test, puis l'effectuer. OWAMP ("*One-way Active Measurement Protocol*") comprend deux parties :

- OWAMP-Control est le protocole bi-directionnel de communication avant le test. Reposant sur TCP (port 861), il sert à se mettre d'accord sur les détails du test.

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc5357.txt>

- OWAMP-Test est le protocole de test, bâti sur UDP. Il fonctionne par l'envoi de paquets contenant l'heure de la machine émettrice. La réceptrice peut donc facilement calculer le temps de trajet, avec une résolution qui est celle des horloges utilisées par les deux machines.

OWAMP pose quelques problèmes de sécurité puisque, dans une communication à sens unique, il peut être difficile de s'assurer que les messages ont bien été transmis et pas altérés. Comme certains acteurs auraient intérêt à fausser les mesures, OWAMP-Protocol permet aux deux parties de se mettre d'accord sur des mécanismes d'authentification et de contrôle d'intégrité des paquets, en utilisant la cryptographie.

Deux mises en œuvre d'OWAMP en logiciel libre sont disponibles en <http://e2epi.internet2.edu/owamp/> et <http://www.av.it.pt/jowamp/> On peut lire aussi "*Comparing 2 implementations of the IETF-IPPM One-Way Delay and Loss Metrics*" http://www.ripe.net/projects/ttm/Documents/Presentations/2000_PAM.pdf, un excellent exposé qui donne beaucoup de détails techniques sur la mise en œuvre de ces mesures de trajets à sens unique. La question de la mesure du temps avec une résolution suffisante (une microseconde) n'est pas évidente sur un Unix de série!

Voici un exemple d'utilisation de l'OWAMP d'Internet2, sur une Debian :

```
# Lancer le démon qui servira pour le protocole de contrôle. Ici, pour les tests,
# on le lance en avant-plan (option -Z)
% sudo owampd -U dupont -Z -v -a 0
...
# Lancer le client (ici, sur la même machine, ::1, l'adresse IPv6 locale)
# et mesurer
% owping ::1
Approximately 13.1 seconds until results available

--- owping statistics from [::1]:43537 to [::1]:43538 ---
SID:      00000001cc734425a191ff7187f24db9
first:    2008-09-11T09:15:50.875
last:     2008-09-11T09:16:01.171
100 sent, 0 lost (0.000%), 0 duplicates
one-way delay min/median/max = 0.0129/0.1/0.0439 ms, (err=1.48 ms)
one-way jitter = 0 ms (P95-P50)
TTL not reported
no reordering
...
```

Pour une critique vigoureuse de ce RFC (pas du protocole, mais de la rédaction du RFC), voir «*How NOT to Write a Protocol Specification*» <http://blog.dustintrammell.com/2008/11/17/how-not-to-write-a-protocol-specification/> », de Dustin D. Trammell.