RFC 8762 : Simple Two-way Active Measurement Protocol

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 20 mars 2020

Date de publication du RFC : Mars 2020

https://www.bortzmeyer.org/8762.html

Ce nouveau RFC décrit un protocole simple pour piloter des mesures comme celle de la latence <https://www.bortzmeyer.org/latence.html> ou comme le taux de pertes de paquets entre deux points. Il fonctionne pour des mesures aller-simple (unidirectionnelles) et aller-retour (bidirectionnelles.)

Mais, vous allez me dire, il existe déjà un tel protocole, TWAMP, normalisé dans le RFC 5357[1], et dans quelques RFC suivants comme le RFC 6038. TWAMP est mis en œuvre dans plusieurs systèmes, et fonctionne. Le problème est que TWAMP est riche et complexe, et que sa simplifiation "*TWAMP Light*" (annexe 1 du RFC 5357) est apparemment insuffisamment spécifiée, menant à des problèmes d'interopérabilité. Voir par exemple le rapport « "*Performance Measurement from IP Edge to Customer Equipment using TWAMP Light*" <https://www.broadband-forum.org/technical/download/TR-390.pdf> ».

Les grandes lignes de STAMP ("*Simple Two-way Active Measurement Protocol*") sont décrites dans la section 3 de notre RFC. Comme TWAMP, STAMP sépare les fonctions de contrôle du test, et le test lui-même. Contrairement à TWAMP, la norme STAMP ne décrit que le test (rappelez-vous que STAMP se veut plus simple que TWAMP). Le contrôle est fait de l'extérieur, par des programmes en ligne de commande, par Netconf ou par un autre moyen.

Avec STAMP, il y a deux acteurs, le "*Session-Sender*" qui envoie les paquets, et le "*Session-Reflector*", qui répond. Les paquets sont de l'UDP vers le port 862, et contiennent un numéro de séquence qui permet d'associer requête et réponse. (La réponse contient davantage de données que la requête, mais la requête est remplie, pour avoir la même taille, afin d'éviter les attaques avec amplification <https://www.bortzmeyer.org/attaques-reflexion.html>.) STAMP a deux modes de fonctionnement (section 4 du RFC) :

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

— Sans état ; le "*Session-Reflector*" ne se souvient pas des numéros de séquence déjà vus, il se contente de répondre à chaque paquet, en recopiant le numéro dans la réponse.
— Avec état ; le "*Session-Reflector*" se souvient des numéros de séquence. Ce mode permet, par exemple, de calculer le taux de perte de paquets (en voyant les trous dans la séquence) dans chaque direction, alors que le mode sans état est limité au bidirectionnel (trajet aller/retour, sans pouvoir distinguer la contribution de l'aller et celle du retour.)

Le paquet STAMP inclus, outre le numéro de séquence déjà cité, une estampille temporelle, par défaut au format utilisé par NTP (cf. RFC 5905, section 6.) Un accord entre les deux parties (souvenez-vous que STAMP ne décrit pas comment l'envoyeur et le réflecteur sont coordonnés) permet d'utiliser d'autres formats comme celui de PTP (RFC 8186.)

L'authentification n'est pas indispensable et, si on ne l'utilise pas, le paquet peut être modifié par un attaquant actif qui cherche à fausser les mesures. STAMP a également une possibilité d'authentification, où le paquet contient en outre un HMAC (RFC 2104, et section 4.4 de notre RFC.)

Dans sa réponse, le "*Session-Reflector*" envoie un paquet qui indique son propre numéro de séquence, sa propre estampille temporelle, et les numéros de séquence et estampille temporelle du paquet reçu. (En outre, dans le mode avec état, le "*Session-Reflector*" enregistre les informations du paquet entrant.) Au retour, le "*Session-Sender*" peut donc calculer le RTT, la gigue et le taux de pertes. (Dans le mode avec état, le "*Session-Reflector*" peut lui aussi calculer ces informations, qui auront l'avantage de ne concerner qu'une seule direction des paquets.)

Notez que STAMP n'a aucune confidentialité. Si on veut empêcher un méchant de lire les informations, il faut emballer le tout, par exemple dans IPsec ou DTLS. Ce point est un de ceux qui avaient suscité les plus vives discussions à l'IETF, certains regrettant qu'il n'y ait pas de sécurité standard.

Notez également qu STAMP et TWAMP-Light peuvent partiellement interopérer, puisque le format de paquets et la sémantique sont les mêmes (en non authentifié, uniquement.)

Je n'ai pas trouvé de mise en œuvre de STAMP, ni en logiciel libre, ni en privateur. Je ne connais pas non plus de liste de serveurs STAMP publics.