RFC 5128 : State of Peer-to-Peer(P2P) Communication Across Network Address Translators(NATs)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 30 mars 2008

Date de publication du RFC : Mars 2008

https://www.bortzmeyer.org/5128.html

Pour faire face à la plaie du NAT, les applications pair-à-pair ont développé de nombreuses techniques pour pouvoir passer à travers les routeurs NAT. Ce RFC les documente.

Le groupe de travail Behave <https://www.bortzmeyer.org/behave-wg.html> de l'IETF travaille à normaliser les techniques permettant de contourner les NAT et les nombreux problèmes qui leur sont associés. Quoique certains à l'IETF auraient préféré qu'aucun travail permettant de rendre les NAT plus supportables ne soit effectué, le consensus a finalement été qu'il valait mieux travailler à faire fonctionner les applications malgré les NAT, plutôt que de rêver supprimer ceux-ci.

Ce RFC ne normalise pas un protocole mais examine l'état de l'art, en matière de fonctionnement en présence de NAT. Il rappelle que les NAT n'apportent rien en matière de sécurité mais, comme cette technique est largement déployée, il est préférable de documenter les bricolages divers qu'utilisent les applications forcées de vivre derrière un NAT.

Il existe deux grandes catégories de techniques pare-NAT. Celles qui reposent sur une signalisation explicite de l'application au routeur NAT. La plus connue est UPNP. L'IETF a son MIDCOM (cf. RFC 4097 [1]) pour le même rôle. Ces techniques ne sont pas étudiées dans notre RFC.

Les autres techniques pare-NAT sont celles qui ne nécessitent pas forcément de communication explicite entre l'application et le routeur NAT. Chacune est détaillée dans une section de ce RFC.

La terminologie utilisée est celle du RFC 2663, pas celle des « cônes » du RFC 3489 et la section 2 est entièrement occupée à débroussailler les questions de vocabulaire.

Les techniques décrites ensuite sont :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc4097.txt

— Le relayage, où une machine située à l'extérieur relaie les paquets entre les deux pairs coincés derrière leur NAT. Très consommatrice de ressources pour le relais, cette technique n'est en général qu'un dernier recours pour les applications P2P. (Le protocole TURN, RFC 5766, est un exemple de cette technique.)
— Le retournement où un des deux pairs peut demander à l'autre d'être l'initiateur de toutes les connexions. Cela n'a d'intérêt que si un des deux n'est pas derrière un NAT.
— Le perçage, ainsi nommé parce que les applications tentent de percer un trou dans le routeur NAT pour laisser les paquets passer. Le principe est d'envoyer un paquet vers le pair, créant ainsi une entrée dans la table du routeur NAT, pour un certain port. Des paquets en sens inverse qui connaissent le port (par exemple parce qu'ils ont contacté un serveur de rendez-vous STUN) peuvent alors y écrire.
— La prédiction de ports UDP ou TCP, très difficile, car elle repose sur des heuristiques sur le prochain numéro de port que choisira le routeur NAT pour sa table.

La section 5 résume ces techniques et les recommandations des auteurs. Le perçage est la technique la plus efficace à l'heure actuelle.

La section 6, consacrée à la sécurité, est très détaillée car les NAT crée de nombreux problèmes de sécurité.

Notre RFC contient aussi, en section 4, une bonne liste des articles sur le sujet, afin d'aider l'auteur d'applications qui veut maximiser les chances de fonctionnement dans toutes les circonstances.